

VTAM™

Network Implementation Guide

Version 4 Release 2 for MVS, VM, and VSE



VTAM™

Network Implementation Guide

Version 4 Release 2 for MVS, VM, and VSE



Note!

Before using this document, read the general information under "Notices" on page xxvii.

Second Edition (February 1995)

This edition applies to the Advanced Communications Function for Virtual Telecommunications Access Method (ACF/VTAM), an IBM licensed program:

VTAM Version 4 Release 2 for MVS/ESA (program number 5695-117)
VTAM Version 4 Release 2 for VM/ESA (program number 5654-010)
VTAM Version 4 Release 2 for VSE/ESA (program number 5686-065).

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to:

IBM Corporation
Department E15
P.O. Box 12195
Research Triangle Park, North Carolina 27709
U.S.A.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1984, 1995. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xxvii
Trademarks	xxvii
About This Book	xxix
Who Should Use This Book	xxix
How to Use This Book	xxix
How This Book Is Organized	xxix
Examples Used in This Book	xxxi
Terms Used in the VTAM Library	xxxi
Symbols Used in the VTAM Library	xxxi
Artwork Used in This Book	xxxii
What Is New in This Book	xxxiii
Installation	xxxiii
LU 6.2 Application Program Interface (API) Enhancements	xxxiii
Network Dynamics	xxxiii
Operation Interface Enhancements	xxxiv
Performance Enhancements	xxxiv
Problem Diagnosis	xxxv
System and Configuration Management	xxxv
Where to Find Information about VTAM	xxxvi

Part 1. Concepts and Installation	1
Chapter 1. VTAM Networking Concepts	3
VTAM Functions	3
VTAM Nodes	3
Nodes with Subarea Function Only	3
System Services Control Point (SSCP)	4
CMCs and Data Hosts	4
Nodes with APPN Function Only	4
Control Point (CP)	5
Network Node	5
End Node	6
Network Node Server	7
Central Directory Server	7
Nodes with Both Subarea and APPN Function	8
Interchange Node	8
Migration Data Host	8
Composite Network Node	9
LEN Node	9
Network Accessible Units	10
Physical Unit (PU)	10
Logical Unit (LU)	10
Dependent LU Support	10
Independent LU Support	11
Network Control Sessions	11
SSCP-SSCP Sessions	11
CP-CP Sessions	12
End-User (LU-LU) Sessions	13

How VTAM Locates Resources	14
Locating Resources in a Subarea Network	14
Locating Resources in an APPN Network	14
Route Selection	15
Routing in a Subarea Network	15
Routing in an APPN Network	16
Topology and Routing Services	16
Topology Database	16
APPN Class of Service	17
Route Calculation and Selection	18
Addressing	18
Subarea Addressing	18
APPN Addressing	19
 Chapter 2. Installing VTAM under MVS	 21
Calculating VTAM Virtual Storage Requirements	21
Preparing Your Operating System	21
Defining Data Sets for VTAM	21
Key Data Sets Containing Information for VTAM	24
Data Sets Containing Information for NCP	29
Defining VTAM to MVS	31
Loading VTAM	34
Starting VTAM	34
Installing the VTAM Dump Analysis Enhancements and the VIT Analysis Tool	35
Concatenating the Target Data Sets Used in the Installation	36
Installing the ISPF Tables	38
Invoking the ISPF Dialog Tag Language Utility	39
Compiling the Help Panels, Keylists, and Commands	40
Verifying the Dump Formatter Panels	42
Verifying the Trace Formatter Panels	43
Customizing the IPCS Interface	45
Customizing the ISPF Interface	46
Using the Online Message Facility	48
Installing the VTAM Command Set Library for IBM Command Tree/2	48
 Chapter 3. Installing VTAM under VM	 51
Calculating VTAM Virtual Storage Requirements	51
Deciding Where to Run VSCS	51
Determining Storage Requirements	52
Storage Layout	52
Preparing Your Operating System	55
5654010A Directory Entry	55
VTAM Directory Entry	57
Enabling Directory Changes	58
Checkpointing APPN Data Files	58
Profile Requirements for Automatic Logon (AUTOLOG)	60
PROFILE for the AUTOLOG1 Virtual Machine	60
PROFILE for the GCS Recovery Virtual Machine	60
PROFILE for the VTAM Virtual Machine	61
Discontiguous Shared Segment (DCSS) and Named Saved Systems	63
Loading VTAM - Overview	64
Installation Instructions	66
Installation of VTAM V4R2 for VM/ESA with VMSES/E (VMFINS)	66
VMSES/E Installation Process Overview	66

Plan Your Installation	67
Allocate Resources for Installing VTAM V4R2 for VM/ESA	71
Install VTAM V4R2 for VM/ESA	71
Update Build Status Table for VTAM V4R2 for VM/ESA	74
Place VTAM into Production	76
Define and Build the VTAM V4R2 for VM/ESA Saved Segments Using VMSES/E	76
Activate the Program	81
Copy VTAM Files Into Production	81
Install VTAM VIT Analysis Tool	82
Install OS/2 Code	82
Defining Discontiguous Shared Segment (DCSS) and Named Saved Systems	82
Generating Group Control System (GCS)	84
Verifying the Installation	85
Applying Service	86
Installing Tables and Modules in VM	86
VTAM User Modifications during Installation:	86
Service Instructions	88
VMSES/E Service Process Overview	88
Servicing VTAM V4R2 for VM/ESA	89
Place the New VTAM Service into Production	92
Trouble Shooting Installation and Service Problems	94
Communication Manager/2's DLUR Support	96
Installing the VTAM VIT Analysis Tool	96
Accessing the Appropriate Disks	96
Updating and Running the ISPF EXEC	97
Installing the ISPF Trace Tables	98
Invoking the ISPF Dialog Tag Language Utility	99
Compiling the Help Panels, Keylists, and Commands	100
Verifying the Trace Formatter Panels	101
Customizing the ISPF Interface	102
First Failure Support Technology (FFST)	104
Installing the Online Message Facility	104
Installing the VTAM Command Set Library for IBM Command Tree/2	104
Chapter 4. Installing VTAM under VSE	107
About This Chapter	107
Calculating VTAM Virtual Storage Requirements	107
Preparing Your Operating System	108
Defining Files for VTAM	108
VTAM-Related Files	109
Checkpointing APPN Data Files	109
Configuration Restart VSAM Files	111
NCP-Related Files	112
VTAM Trace Files	113
Defining VTAM to VSE	114
SUPVR Macroinstruction	114
Coding an IPL Procedure with Support for VTAM (Optional)	114
Loading VTAM	117
Starting VTAM	117
Keeping an Application Program Active If VTAM Abends	119

Chapter 5. Getting Started with VTAM - Overview	125
Implementing a VTAM Network	125
Using Start Options and Configuration Lists	127
Start Options	127
Required Start Options	127
Recommended Start Options	128
Sources of Start Options	129
Start Option Processing	130
Creating Start Option Lists	130
Improving VTAM Performance Using Start Options	131
Diagnosing Problems Using Start Options	133
Configuration Lists	134
Identifying Resources to VTAM	136
Coding Concepts	136
Sift-Down Effect	137
Initial Status	137
Defining Application Programs	138
Attaching NCP to VTAM	138
Defining Peripheral Nodes and Logical Units	139
Defining Type 2.1 Peripheral Nodes	139
Independent LUs	140
Defining Dependent Logical Units	141
Defining Dependent LUs Dynamically	141
Attaching Peripheral Nodes to VTAM	142
Attaching Peripheral Nodes to NCP	142
Defining Switched Resources Dynamically	143
Attaching Other Domains to VTAM	145
Sharing NCP Peripheral Nodes (CMC Configurations)	145
Dynamic Configuration of Channel-Attached Devices	147
Dynamic Reconfiguration and Change of Operands	149
Identifying Routes and Establishing Sessions	150
Network Routing for Subarea Nodes	151
Establishing and Controlling SNA Sessions	155
Logon and Logoff Requests from Dependent Logical Units	156
National Language Support	157
Model Name Table	157
Associated LU Table	157
Selecting Session Parameters for the Logon Mode Table	158
Automatic Logons	158
Session Authorization and Accounting	158
Pacing and Flow Control	159
Application Program Pacing	159
Setting Pacing Values	159
Operating VTAM	159
Verifying a VTAM Network	160
 Chapter 6. Getting Started with VTAM - Detail	 161
Application Programs	161
Naming an Application Program	161
Authorizing Application Facilities	162
Data Compression	163
Types of Compression	164
Implementing Data Compression	165
Compression Level Negotiation	166

Security Facilities	168
Encryption Facility	168
VTAM Application Security	172
Confidential Data	173
Logon Mode Parameters	173
High Availability Using Extended Recovery Facility	174
Cryptography in an XRF Environment	175
Using User Variables (USERVAR)	175
Application Workload Balancing with USERVAR	176
Classes of USERVARs	176
Types of USERVARs	177
Processing USERVARs	178
USERVAR Propagation and Routing	180
Defining Your Network with USERVARs	180
Dynamic USERVAR Update Session Failure	181
Generic Resources	182
Generic Resources Requirements	183
Generic Resource Mapping	184
Partner LU Mapping	185
Initiating Sessions With Generic Resource Members	187
Implementation Considerations	188
Connecting to and Allocating Storage for ISTGENERIC	189
The Structure Attributes Associated with ISTGENERIC	189
Defining a Structure Size for ISTGENERIC	191
Failure Recovery for the Generic Resource Configuration	191
Structure Rebuild	192
Routine Maintenance for VTAM Nodes	193
LU 6.2 in an XRF Environment	193
Persistent LU-LU Sessions	193
VTAM Common Network Services	195
Cross-Memory Application Program Interface (API)	195
Allocating Private Storage	195
Communicating with Start-Stop Devices	196
LU 6.2 Application Programs	196
Enabling LU 6.2 Support	196
LU 6.2 Sessions	196
LU 6.2 Session Limits	197
Managing LU 6.2 Sessions with Operator Commands	199
LU 6.2 Security	199
LU 6.2 Conversation-Level Verification	200
LU 6.2 Sync Point Services	201
Selective Termination of Idle LU 6.2 Sessions	201
Attaching NCP to VTAM	202
NCP Definition Statements for VTAM	202
PCCU Definition Statement	202
BUILD Definition Statement	204
HOST Definition Statement	206
NCP Definition Statements for NCP	207
Channel-Attached NCP	207
Using Parallel Transmission Groups between VTAM and a	
Channel-Attached NCP	209
Link-Attached NCP	212
Recovery Operations	212
Defining a Link-Attached NCP	212

Enhanced Remote Controller Support	214
NCP Switched SDLC Connections	215
Defining Dependent LUs Dynamically	216
Independent LUs	218
Characteristics of Independent LUs	218
Defining Independent LUs	220
Dynamic Definition of Independent LUs	221
Using CDRSC Definition Statements for Independent LUs	222
Automatic Conversion of Independent LU Definition Statements	223
Multiple Connections between a Type 2.1 Node and a Subarea Node	224
Dynamic Selection of Session Connections	225
Connecting Independent LUs through NCPs	226
Authorized Transmission Priority for LEN Connections	227
Restrictions on Using Independent LUs	227
Attaching Peripheral Nodes to VTAM	228
Channel Connections	228
Local Non-SNA Connection	228
Local SNA Connection	229
Loop-Adapter-Attached Connection	231
External Communication Adapter (XCA) Connections	231
Defining an XCA Configuration	231
Example XCA Configuration	232
Automatically Generating Lines and Physical Units	233
Integrated Communication Adapter (ICA) Connections	234
Nonswitched SDLC Connection	234
Switched SDLC Connection	235
X.21 Switched Line	237
Token-Ring Connection through the ICA	239
X.25 Connection	243
X.25 DTE-to-DTE Support through the ICA	247
BSC Connection	248
Attaching Peripheral Nodes to NCP	249
Defining Type 2.1 Peripheral Nodes	249
Non-Native Network Type 2.1 Connections	250
NCP Connections	250
Nonswitched SDLC Connection	250
Switched SDLC Connection	251
NCP/Token-Ring Interconnection (NTRI)	256
NCP Packet Switching Interface (NPSI) Connections	259
BSC Connection	259
Dynamically Defining Switched Resources	260
Attaching Other Domains to VTAM	262
VTAM-to-VTAM Channel Connections	262
Channel-to-Channel Adapter Connection	262
Channel-Attached NCP Connection	264
Multipath Channel (MPC) Connections	266
VTAM-to-VTAM NCP Connections	267
Nonswitched SDLC Subarea Connection	267
Switched SDLC Subarea Connection	269
X.25 Subarea Connection (NPSI Support)	272
Token-Ring Connection (NTRI Support)	273
VTAM-to-VTAM XCA Connections	274
Sample Configuration with a Token-Ring Local Area Network	274
Sample Configuration with CSMA/CD 802.3 Local Area Network	276

VTAM-to-VTAM ICA Connections	278
Nonswitched SDLC Subarea Connection	278
Switched SDLC Subarea Connection	279
X.25 Subarea Connection	280
DTE-to-DTE Connections	282
Token-Ring Connection	283
Sharing NCP Peripheral Nodes (CMC Configurations)	284
Channel-Attached NCPs in a CMC Configuration	284
Link-Attached NCPs in a CMC Configuration	286
Controlling Resource Ownership in a CMC Configuration	286
Problem Determination in a CMC Configuration	287
Dynamic Configuration of Channel-Attached Devices	287
Installation and Preparation	287
Defining Your Configuration	288
Building Resource Definitions	288
Using the Default Naming Convention	289
Customizing the Command Lists	290
Modifying the Resource Definition Defaults	290
Modifying the Naming Convention	290
Supplying Your Own Naming Convention	291
Writing Your Own Command Lists	291
Dynamic Reconfiguration and Change of Operands	291
Dynamic Reconfiguration and Dynamic Change Requirements	291
Using the VARY ACT,UPDATE Technique	292
Coding Dynamic Reconfiguration Changes	294
Errors in the VTAMLST Definition File	295
Using the VARY DRDS Technique	295
Adding Resources	296
Deleting Resources	297
Moving Resources	297
Using the MODIFY DR Technique	297
Dynamic Reconfiguration of Independent LUs	297
Dynamic Reconfiguration of Frame Relay Resources	298
Implicit Dynamic Reconfiguration	300
Dynamic Reconfiguration Examples	301
Line Backup through Dynamic Reconfiguration	304
Network Routing for Subarea Nodes	306
How VTAM Handles Network and Subarea Addressing	306
Compatibility between Non-Extended and Extended Network Addressing	
Nodes	307
Compatibility between Extended Network and Extended Subarea	
Addressing Nodes	309
Virtual Route Pacing	310
Parallel Sessions Using Parallel Transmission Groups	312
Dynamic Path Update	313
Establishing and Controlling SNA Sessions	315
National Language Support for End-User USS Messages and Commands	316
Defining USS Tables for Use with the LANGTAB Operand	316
Defining USS Messages to the MVS Message Service	317
National Language Support for the Language Passed to Application	
Programs	317
Model Name Table	318
Associated LU Table	319
Selecting Session Parameters for the Logon Mode Table	320

Automatic Logons	323
Coding for Automatic Logon	323
Operator Commands for Automatic Logon	325
Reallocation of Autologon Sessions	326
Session Authorization and Accounting	329
Session Authorization	329
Session Accounting	330
Pacing and Flow Control	330
Session Pacing	330
Fixed Session Pacing	332
Overriding Fixed Pacing Counts	333
Adaptive Session Pacing	334
Application Program Pacing	337
Inbound Session Pacing	337
Outbound Session Pacing	338
Setting Pacing Values	339
Operating VTAM	341
Starting the Domain	341
Configuration Restart	342
Information Recorded by Configuration Restart	347
Activating Resources	348
Order of Activation	348
Activating Application Programs	349
Monitoring the Domain	350
Using the DISPLAY Command	350
Monitoring I/O Problems	350
Suppressing Messages	351
Message Flooding Prevention	351
Using the Online Message Facility	352
Displaying and Testing Routes	352
Defining Operator Messages and Commands	352
Multiple Console Support (MCS) in VTAM	353
Controlling the Domain	354
Establishing and Terminating Sessions with Operator Commands	354
Controlling X.25 Connections	354
Dynamic Table Replacement	354
Deactivating Resources	355
Order of Deactivation	355
Automatic Deactivation	355
Normal Deactivation	355
Immediate Deactivation	356
Forced Deactivation	356
Forced Reactivation	356
Halting VTAM	356
Canceling VTAM	358
Automatic Operations	358
Program Operators	358
CNM Application Programs	359
Reducing Session Awareness (SAW) Data	360
Operating with an NCP	361
Loading an NCP	362
NCP Load Process	362
Identifying the Communication Controller	363
Selecting a Load Station	364

Loading from the Host or from External Storage	365
Reloading an NCP	365
Scheduled Automatic Reloading of a Communication Controller	366
Renaming NCP Load Modules	368
Loading or Purging a Load Module on a Communication Controller Disk	370
Activating an NCP	371
Automatic Activation of Adjacent Link Stations	371
Reactivating a Failed NCP	374
Activating Links	375
Restrictions on Activation	376
Loading and Activating an NCP over a Switched Link	377
Enabling Communication Controller Channels for Activation	378
Monitoring an NCP	379
Sending Line Errors to VTAM	379
Displaying NCP Storage	380
Testing a Nonswitched SDLC Line	380
Deactivating an NCP	381
Deactivating Links	381
Dumping an NCP	382
Selecting a Dump Station for a Static Dump of an NCP	384
Selecting a Dump File for an NCP Dump	385
Selecting the Disposition of an NCP Dump	385
Other Communication Controller Dumps	385
Selecting a Dump File for a MOSS or CSP Dump	386

Part 3. Subarea Networks and SNA Network Interconnection (SNI) 387

Chapter 7. Implementing a Subarea Network	389
Implementing a Subarea Network	389
Starting VTAM	391
Defining the Location of Cross-Domain Resource Managers by Coding	
Adjacent SSCP Tables	391
Defining the Location of Cross-Domain Resource Managers Dynamically	392
Specifying Time-Out Values for Locating Cross-Domain Resources	392
Identifying VTAMs in Other Domains (CDRMs)	393
Identifying Resources in Other Domains	395
Dynamic Definition of Cross-Domain Resources	395
Static Definition of Cross-Domain Resources	397
Adjacent SSCPs	398
Adjacent SSCP Lists for CDRSCs	400
Defining Adjacent SSCP Tables	400
Table Search Order	402
Routing with Adjacent SSCP Tables	403
Activating Adjacent SSCP Tables	404
Improving Performance	404
CDRM Owner Verification for Cross-Domain Resources	406
Changing Ownership of Cross-Domain Resources	406
Shadow Resources	407
Type 2.1 Casual Connection	407
Defining a Type 2.1 Casual Connection	408
Considerations for Using Casual Connection	413
Routing in a Subarea Network	414
How Session Traffic Is Assigned to a Specific Route	414

How to Plan Routes in Your Network	418
Subarea Addressing	422
Compatibility between Non-Extended and Extended Network Addressing	
Nodes	423
Compatibility between Extended Network and Extended Subarea	
Addressing Nodes	425
Establishing and Controlling SNA Sessions	427
Operating VTAM in a Multiple-Domain Subarea Network	427
Activating Resources	427
Deactivating Resources	428
Operating with an NCP	428
Links and Link Stations	429
Activating Links	429
Deactivating Links	429
Cross-Domain Links	430
Delayed Activation of Logical Lines	430
Discontiguous Domains	432
Backing Up Resource Owners	432
Steps for Resource Takeover	432
Restarting the Host	433
Returning Ownership	434
Sharing an NCP	434
Verifying a Multiple-Domain Subarea Network	435
 Chapter 8. Connecting Multiple Subarea Networks	 437
Implementing a Multiple-Network Environment	437
SNA Network Interconnection	437
Defining Your Network	437
SNI Configurations	440
Starting VTAM	441
Start Options for Gateway VTAMs	441
GWSSCP Start Option for Non-Gateway VTAMs	442
Configuration Lists for Gateway VTAMs	442
Application Programs	442
Defining the NetView Program	442
Connecting Networks	443
Defining a Gateway VTAM	445
Defining a Gateway NCP	446
Converting a Non-SNI NCP to a Gateway NCP	447
Defining Native Networks	447
Coding the BUILD Definition Statement	447
Coding the GWNAU Definition Statement	448
Coding the HOST Definition Statement	449
Defining Non-Native Networks	450
Dynamic Network Identification	451
Enabling Dynamic NETID	452
Creating Dynamic Networks from Model Networks	452
Considerations for Implementing Dynamic Network Identifier	452
Sample Dynamic NETID Configuration	453
Defining Cross-Domain Resource Managers	455
Defining Cross-Domain Resources	458
Session Initiation Request	459
Name Assumption	460
Predefined Cross-Domain Resources without Network Specification	462

Predefined Cross-Domain Resources with Network Specification	463
Dynamic Cross-Domain Resources	465
Defining Adjacent SSCP Tables	467
Types of Adjacent SSCP Tables	467
Deciding Whether to Code Adjacent SSCP Tables	468
Sample of Adjacent SSCP Tables for a Multiple-Network Environment . .	468
Example Default Lists	470
Example Network-Specific Lists	470
Overriding the SSCP Lists	471
Request Routing	471
Rerouting of Requests	471
Rules for Cross-Network Session Request Routing	472
Dynamically Defined CDRSCs and Adjacent SSCP Tables	472
Alias Name Translation and Adjacent SSCP Tables	473
Cross-Network Routing	473
Network Address Structures	473
Maintaining Address Structures	473
Addressing Considerations for Interconnected Networks	474
Network Naming Conventions	474
Controlling Paths for Interconnected Networks	475
Gateway Path Selection	475
Coding the Gateway Path	477
Handling Class-of-Service Tables	477
Address Translation	479
Element Addresses	480
Parallel Sessions and Element Addressing	481
Multiple ACBs and Element Addressing	481
Resource Name Translation	481
Alias Selection Function of the Session Management Exit Routine	482
NetView Alias Name Translation Facility	483
Translating LU Names	483
Mapping Logon Mode Names	484
Mapping Class-of-Service Names	484
Defining Alias Names	485
Resolving Alias Name Requirements	488
Establishing and Controlling SNA Sessions	490
Non-Native Network Type 2.1 Connections	490
Considerations for Network Nodes	490
Considerations for End Nodes	490
Specifying and Defining the Destination LU	491
Automatic Logon	491
Session Management Exit Routine	492
Session Accounting	492
Operating VTAM	493
Activating Resources	493
Gateway NCP and VTAM interaction	493
Terminating Sessions	493
Using the NetView Program for Network Management	493
Verifying a Multiple-Network Environment	494

Part 4. Implementing APPN in Your Network	495
--	------------

Chapter 9. Implementing a Combined APPN and Subarea Network	497
--	------------

Implementing a Combined APPN and Subarea Network	497
Starting VTAM	501
SORDER Start Option	501
SSEARCH Start Option	502
CDRSCTI Start Option	502
Dependent LUs	502
Dependent LU Server	502
Dependent LU Server Support Across Subnetwork Boundaries	506
Defining CDRSCs	508
Shared NCPs	508
SSCP Takeover	509
Establishing and Controlling Sessions	512
Controlling Searches	512
Using the SORDER Start Option to Control Network Search Order	512
Using the SSEARCH Start Option to Limit Subarea Network Searches	513
Using the CDRSCTI Start Option to Reduce Broadcast Searches of APPN	513
Using the DISJOINT Operand to Define Disjoint SNI Networks	513
Virtual-Route-Based Transmission Groups (VR-Based TGs)	515
Defining a VR-based TG	517
VR-Based TG Recommendations	518
APPN and Subarea Class of Service Resolution	520
Adding and Moving Nodes	522
Verifying a Combined APPN and Subarea Network	523
 Chapter 10. Implementing an APPN Network	 525
Implementing an APPN Network	525
Starting VTAM with APPN Support	525
Enabling Control Sessions	525
CP-CP Sessions Between Two VTAM Nodes	526
Defining Adjacent APPN Nodes	526
Defining the Logon Mode for CP-CP Sessions	526
Connecting an APPN Node to the Network	527
Connections through Boundary Function-Based Transmission Groups	527
Multiple Connections with Parallel Transmission Groups	528
Connection Network	528
Defining an APPN Connection	533
Using Type 2.1 Channel Connections between APPN Nodes	535
Using Leased Connections between APPN Nodes	537
Using XCA over a 3172 Interconnect Controller between APPN Nodes	539
Using ICA over X.25 between APPN Nodes	540
Using ICA over a Token Ring between APPN Nodes	542
Using ICA over an SDLC Connection between APPN Nodes	543
Defining a Network Node Server List	544
Creating a Network Node Server List	545
Activating, Replacing, and Displaying a Network Node Server List	545
APPN Multiple Network Connectivity	545
Peripheral Subnetwork Boundaries	548
Extended Subnetwork Boundaries	548
Using APPN Multiple Network Connectivity Support	549
Network Routing and Resource Location for APPN Nodes	550
Types of Searches	551
Directed Search	551
Broadcast Search	552
Cross-Subnetwork Search	553

Minimizing Broadcast Searches to Improve Performance	553
Central Directory Servers	553
Registering Resources	554
Controlling the Timeliness and Size of the Directory Database	555
Checkpointing of the TRS Database and the Directory Database	555
APPN Class of Service	556
What is APPN class of service?	556
What information is specified in the APPN class of service?	556
Do you have to code anything to make APPN class of service work?	556
What are the IBM-supplied default classes of service?	557
What do you get if you take all the defaults?	557
What is the APPN class of service default in a user-coded logon mode table?	558
How do I add APPN class of service to my logon mode table?	558
How does VTAM use the class of service to choose a route?	558
Influencing Session Routing	559
Example of Weight Computation	559
Verifying a VTAM APPN Network	560

Part 5. Tuning VTAM for Your Environment 563

Chapter 11. Tuning VTAM for Your Environment	565
Introduction to Tuning	565
Tuning Tools	566
Estimating Active Sessions	566
Common Storage Areas	566
Buffer Pools	566
NCP Slowdown	567
Coattailing	567
Tuning Tools	567
Monitoring Common Storage Areas	567
Analyzing Slowdown	569
Gathering Tuning Statistics	569
Tuning Input/Output (I/O) Operations for SNA Controllers	570
Tuning I/O Operations for Channel-to-Channel Connections	571
Tuning I/O Operations for Multipath Channel Connections	574
Analyzing Tuning Statistics	576
Determining the Amount of Coattailing in Your System	577
Migrating from User-Replaceable Constants	577
Estimating the Number of Active Sessions	577
EAS Operand for Application Programs	578
EAS Operand for Independent Logical Units	578
Common Storage Areas	578
Common Storage Areas for MVS and VM	578
Common Service Area Limit	578
Common Service Area 24-Bit	579
Common Storage Areas for VSE	579
System GETVIS Area	579
System GETVIS Area 24-Bit	579
DISPLAY STORUSE Pools	579
Buffer Pools	583
Types of Buffer Pools	583
Buffer Pool Allocation	585

Setting Buffer Pool Allocations	585
How Dynamic Expansion Operates	586
Guidelines for Dynamic Expansion	587
Dynamic Expansion and Storage	591
I/O Buffers and Application Program Data Transfer	591
HOT I/O Detection/Termination	592
HOT I/O Session Detection	593
HOT I/O Session Termination	594
Controlling NCP Slowdown	594
Maximizing Coattailing	595
Controlling Outbound Coattailing	595
Controlling Inbound Coattailing	596
DELAY Operand	596
MAXBFRU Operand	597
Coattailing for SNA Controllers	597
Guidelines for Setting DELAY	598
Guidelines for Setting MAXBFRU	600
Guidelines for Setting I/O Buffer Size	601
Guidelines for Setting UNITSZ	603
Coattailing for Channel-to-Channel Operations	603
Guidelines for Setting DELAY	603
Guidelines for Setting I/O Buffer Size	604
Virtual Route Window Sizes	604

Appendixes	607
-------------------	-----

Appendix A. TSO/VTAM	611
Defining TSO to VTAM	611
Defining the TCAS Application to VTAM	612
Single-Domain Network	612
Multiple-Domain Network	613
Resource Registration in an APPN Network	614
Defining TSO/VTAM Session Parameters	614
TSO/VTAM Support of 3270 Devices	615
Defining the 3790/3270 Configuration to TSO/VTAM	618
Defining 2741, TWX, or WTTY Terminals to TSO/VTAM	619
Defining an Interpret Table for Compatible Logons	620
Defining TSO to MVS	620
Writing a Procedure to Start TSO/VTAM Time Sharing	620
Creating a TSOKEY00 SYS1.PARMLIB Member	620
Defining TCAS Program Properties	620
Implementing TSO/VTAM	621
Translation Tables	621
Coding TSO/VTAM Exit Routines	621
Security	621
Performance	622
3270 Large Screen Considerations	622
TSO/VTAM Screen Management	622
Full-Screen Application Program Screen Management	622
TSO Considerations	623
National Language Support for TSO/VTAM End-User Messages	623
Operating VTAM under TSO	624
How TSO/VTAM Differs from TSO through TCAM	624

Appendix B. VM/SNA Console Support	627
Preparing to Start VSCS	627
Defining VSCS as a VTAM Application Program	627
Authorizing the VSCS Virtual Machine	628
Updating the Directory Entry for the VSCS Virtual Machine	628
Preparing the GCS Environment to Start VSCS	629
Tailoring VSCS	630
Changing VSCS Start Options	630
Session Parameters for Devices: Logon Mode Tables	631
LOGON and LOGOFF Commands: USS Tables	632
Data Streams: VSCS Data Manipulation Exit Routines	632
Protecting Confidential Data	633
VSCS Start Options	633
Controlling VSCS Storage Management	633
Controlling VSCS Dumps	635
Controlling VSCS Performance	636
Controlling VSCS SCIP Recovery	638
Defining Your Own Set of VSCS Start Options	639
Using VSCS Diagnostic Tools	640
Tracing Commands	640
Display Command	641
LU Dump Command	641
Storage Command	641
Buffer Commands	641
Query Command	641
Halting VSCS	642
Additional VSCS Considerations	642
Printers	642
Terminal Logons	643
Terminal Logoffs	644
VM Logo Usage	645
Accounting Information	647
 Appendix C. Tuning VM for VTAM	649
SET RESERVE	649
DIAG98	649
Page Locking	649
XEDIT SET REMOTE	650
SET QUICKDSP	650
SET SHARE RELATIVE 10000	651
 Appendix D. Logon Manager	653
How the Logon Manager Operates	653
Installing the Logon Manager	654
Starting the Logon Manager	654
Defining the Logon Manager	655
Sample Logon Manager Configuration	655
Defining the Logon Manager and TPF Applications to VTAM	656
Defining the Logon Manager Configuration	657
Subarea Configuration	657
Class-of-Service Configuration	657
Application Passwords and Minimum Channel Links	659
Logon Manager Procedure	659
Number of TPF Applications	660

Maximum Number of Buffers	660
Maximum Number of Subareas	660
Reserved Keywords	660
Monitoring Logon Manager Resources	661
Halting the Logon Manager	661
Appendix E. Cryptographic Keys	663
Filing SLU Keys for Single-Domain Cryptographic Sessions	663
Single-Domain Cryptographic Sessions That Use PCF/CUSP	663
Single-Domain Cryptographic Sessions That Use ICSF/MVS	664
Filing CDRM Keys for Cross-Domain Cryptographic Sessions	665
Cross-Domain Cryptographic Sessions Where Both Hosts Use PCF/CUSP	665
Cross-Domain Cryptographic Sessions Where Both Hosts Use ICSF/MVS	667
Cross-Domain Cryptographic Sessions Where the Hosts Use Different Cryptographic Services	669
Changing a Logical Unit's Cryptographic Capability	671
Appendix F. Command Lists: Dynamic Configuration of Channel-Attached Devices	673
ISTDINFO: VTAM Device Information Services	673
Dependencies and Restrictions	673
Output Variable	674
Output Tokens	674
ISTDEFIN: VTAM Device Information Services	677
Dependencies and Restrictions	677
Output Variable	678
Output Tokens	678
Appendix G. Message Translation Using the MVS Message Service	681
Overview of MMS Support	681
Internal Translation	681
Selecting Internal Translation	681
Defining Messages for Internal Translation	682
External Translation	682
Selecting External Translation	682
Defining Messages for External Translation	683
Skeleton File Use	684
Appendix H. Forcing an APPN Route in a VTAM Network	685
Appendix I. Border Node Connection Types	691
Appendix J. VTAM Restricted Materials	695
Appendix K. VTAM AnyNet Feature	697
Overview of Multiprotocol Transport Networking (MPTN)	697
Peer and Transport Relationships	697
MPTN Architecture Functions	698
Compensation	698
Address Mapping	698
Network Interconnection	698
Network Management	699
MPTN Transport Services	699
Connection-Oriented Services	699

Connectionless Services	701
Where to Use MPTN Connections	702
MPTN Network Management	702
Sockets over SNA Network Management	703
Overview of the VTAM AnyNet Feature	703
SNA over TCP/IP	703
Sockets over SNA	704
Sockets over SNA Gateway	704
Appendix L. Communications Manager/2's DLUR Support	705
Downloading the DLUR Files	705
Unpacking the DLUR Files into Communications Manager/2 Subdirectories	706
Considerations for VTAM Switched Major Node Definitions for DLUR	708
Additions to Communications Manager/2 Configuration Commands for DLUR	708
Addition to DEFINE_LOGICAL_LINK	709
DEFINE_DEPENDENT_LU_SERVER	709
Full Syntax	709
Operand Descriptions	709
Sample DEFINE_DEPENDENT_LU_SERVER	711
Sample Communications Manager/2 NDF File for DLUR	711
Sample Communications Manager/2 RSP File for DLUR	712

Glossary, Bibliography, and Index	723
--	------------

Glossary	725
Bibliography	727
VTAM V4R2 Publications	727
Softcopy Information	727
IBM Networking Softcopy Collection Kit CD-ROM (SK2T-6012)	727
Marketing Information	727
Planning	727
Planning for NetView, NCP, and VTAM (SC31-7122)	727
Planning for Integrated Networks (SC31-7123)	727
Planning Aids: Pre-Installation Planning Checklist for NetView, NCP, and VTAM (SX75-0092)	727
VTAM Licensed Program Specifications (GC31-6490)	727
VTAM Release Guide for MVS/ESA (GC31-6492)	728
VTAM Release Guide for VM/ESA (GC31-8089)	728
VTAM Release Guide for VSE/ESA (GC31-8090)	728
VTAM Migration Guide for MVS/ESA (GC31-6491)	728
VTAM Migration Guide for VM/ESA (GC31-8071)	728
VTAM Migration Guide for VSE/ESA (GC31-8072)	728
Estimating Storage for VTAM (SK2T-2007)	729
VTAM Overview for VM/ESA and VSE/ESA (GC31-8114)	729
VTAM Glossary (GC31-6558)	729
Installation, Resource Definition, and Tuning	729
VTAM Network Implementation Guide (SC31-6494)	729
VTAM Resource Definition Reference (SC31-6498)	729
VTAM Resource Definition Samples (SC31-6499)	729
Operation	729
VTAM Operation (SC31-6495)	729
VTAM Operation Quick Reference (SX75-0205)	729

VTAM Messages and Codes (SC31-6493)	730
Using IBM CommandTree/2 (SC31-7013)	730
Customization	730
VTAM Customization (LY43-0063)	730
Writing Application Programs	731
VTAM Programming (SC31-6496)	731
VTAM Programming for LU 6.2 (SC31-6497)	731
Diagnosis	731
VTAM Diagnosis (LY43-0065)	731
VTAM Data Areas for MVS/ESA (LY43-0064)	731
VTAM Data Areas for VM/ESA (LY43-0103)	731
VTAM Data Areas for VSE/ESA (LY43-0104)	731
VTAM AnyNet Feature for V4R2 for MVS/ESA	732
VTAM AnyNet Feature for V4R2: Guide to Sockets over SNA Gateway for OS/2 (SC31-6528)	732
VTAM AnyNet Feature for V4R2: Guide to Sockets over SNA (SC31-6526)	732
VTAM AnyNet Feature for V4R2: Guide to SNA over TCP/IP (SC31-6527)	732
MPTN Architecture Publications	732
APPC Application Suite Feature for V4R2 for MVS/ESA	732
APPC Application Suite User's Guide (SC31-6532)	732
APPC Application Suite Administration (SC31-6533)	732
APPC Application Suite Programming (SC31-6534)	732
Related Publications	732
Index	733

Figures

1.	Conventions Used in Network Illustrations	xxxii
2.	Correlation between DD Statement and NCP Definition Statement	30
3.	Example of a LOGON PROC	37
4.	ISPF Panel Used to Run the Table EXECs	39
5.	ISPF Panel Used to Invoke the Dialog Tag Language Utility	40
6.	Compiling the Help Panels, Keylists, and Commands for Formatted Dump	41
7.	Compiling the Help Panels, Keylists, and Commands for Formatted Trace	42
8.	Messages Resulting from Compilation	42
9.	Verifying the Dump Formatter Panels	43
10.	Main Menu for Selecting Dump Options	43
11.	Verifying the Trace Formatter Panels	44
12.	Main Menu for Selecting Trace Parameters	44
13.	Sample IPCS Panel BLSPPRIM Customization	45
14.	Addition of Option 7 on the IPCS Primary Option Menu	46
15.	Sample ISPF Panel ISR@PRIM Customization	47
16.	Addition of Option V on the ISPF/PDF Primary Option Menu	48
17.	Sample Storage Layout for VM/ESA VTAM Virtual Machine	54
18.	Installing and Maintaining VTAM under VM	65
19.	Segment Map panel example.	77
20.	Add Segment Definition Panel Example	78
21.	Add Segment Definition Panel Showing the Segments	78
22.	Segment Map panel	79
23.	Sample ISPF EXEC Modified for IST Maclibs	98
24.	ISPF Panel Used to Run the Table EXECs	99
25.	ISPF Panel Used to Invoke the Dialog Tag Language Utility	100
26.	Compiling the Help Panels, Keylists, and Commands for Formatted Trace	101
27.	Verifying the Trace Formatter Panels	101
28.	Main Menu for Selecting Trace Parameters	102
29.	Sample ISPF Panel ISR@PRIM Customization	103
30.	Addition of Option V on the ISPF/PDF Primary Option Menu	104
31.	VSE Storage Layout	108
32.	Sample Definition of APPN Checkpointing Data Files	110
33.	VTAM Network	126
34.	Configuration List	135
35.	Creating Resources in Dynamic Switched Major Node	144
36.	Example of Communication Management Configuration	146
37.	Dynamic Configuration of Channel-Attached Device	148
38.	One Explicit Route in Each Direction	153
39.	Two Explicit Routes in Each Direction	153
40.	Virtual Route and Explicit Route Associations and Transmission Priority	155
41.	Example Session Flow for Compression-Level Negotiation	166
42.	Data Compression Yield	167
43.	Encryption Facility in Single-Domain Environment	169
44.	Encryption Facility in Multiple-Network Environment	170
45.	Encryption Facility in an APPN Environment	172
46.	Sample Extended Recovery Facility Network	178
47.	Generic Resource Mapping	184
48.	Partner LU Mapping	186
49.	Session Establishment with Generic Resource Members	187

50.	Application Program Backup Using Persistent LU-LU Sessions	194
51.	Channel-Attached 3720 Communication Controller	208
52.	Parallel Transmission Groups in Single-Domain Environment	211
53.	Defining, Loading, and Activating a Link-Attached NCP	212
54.	Definition Building for Dynamically Defined Dependent LUs	217
55.	Independent LU with Multiple Connections to VTAM	224
56.	Local SNA Devices	230
57.	XCA Connection in a Single-Domain Environment	232
58.	IBM Token-Ring Subsystem Controller and PORT Definition Statement Operands	240
59.	Multiple Subsystems in ES/9000 Series Processor	242
60.	X.25 Based Configuration in Single-Domain Environment	245
61.	DTE-to-DTE Configuration in Single-Domain Environment	248
62.	Non-Native Network Type 2.1 Connection	250
63.	Major Elements of PATH Definition Statements for Switched Major Node	255
64.	Parallel Transmission Groups in Multiple-Domain Environment with NCP	264
65.	Nonswitched SDLC Subarea Connection	268
66.	Enhanced Remote Controller Support	272
67.	Attaching Domains Using NCP-to-NCP Token-Ring Connection	273
68.	XCA Multiple-Domain Configuration	275
69.	XCA Multiple-Domain Configuration with CSMA/CD	276
70.	Parallel Transmission Groups in Multiple-Domain Environment with ES/9000 Series Processor	278
71.	X.25 Configuration in Multiple-Domain Environment	280
72.	DTE-to-DTE Multiple-Domain Configuration	283
73.	DR ADD Operation	301
74.	DR DELETE Operation	302
75.	DR MOVE Operation	303
76.	Sample Implicit Dynamic Reconfiguration	304
77.	Line Backup through Dynamic Reconfiguration	305
78.	Element and Subarea Address Incompatibility in Single-Domain Environment	308
79.	Local Flow Control	312
80.	Parallel Sessions Using Parallel Transmission Groups	313
81.	Sample Single-Domain Network	315
82.	Macroinstructions for Logon Mode Table	320
83.	How Session Parameters Are Obtained from a Logon Mode Table	324
84.	Automatic Logon to A50ACCTS Application Program	325
85.	Automatic Logon Reallocation	327
86.	One- and Two-Stage Pacing	331
87.	Pacing Counts Used to Control One- and Two-Stage Pacing	333
88.	Adaptive Session Pacing Window Mechanism	335
89.	Pacing: Inbound to VTAM When Application Program Is PLU	336
90.	VTAM and NCP Adaptive Session Pacing Window Size	337
91.	VTAM Adaptive Session Pacing Window Size	338
92.	Pacing Windows for SNA LUs	340
93.	Pacing Windows for Non-SNA LUs	341
94.	Restoring Resource Definitions with Configuration Restart and NODELST Files	343
95.	Using Configuration Restart and NODELST Files	345
96.	NCP Generation	363
97.	Activating NCPs and Link Stations: Through Another NCP	372
98.	Activating NCPs and Link Stations: Through Host Link Station	373
99.	Automatic Activation of Link Stations	373

100. Loading and Activating NCP Using Switched Link	378
101. Loading over a Channel	379
102. Multiple-Domain Network	390
103. Major and Minor Nodes in Multiple-Domain Environment	393
104. Example of Adjacent SSCP Table Connection	402
105. NCP Type 2.1 Casual Connection	412
106. Class-of-Service Substitution	417
107. Class-of-Service Hierarchy	418
108. Element and Subarea Address Incompatibility in Multiple-Domain Environment	425
109. Effects of NCP Deactivation on Cross-Subarea Links and Link Stations	430
110. Multiple-Network Environment	439
111. Backup of a Gateway NCP	454
112. Multiple-Network Configuration: CDRM Major Nodes	457
113. Example of Three Interconnected Networks	460
114. Example of Two Interconnected Networks	462
115. Multiple-Network Configuration	468
116. Multiple-Network Paths	475
117. COS Resolution in a Multiple-Network Environment	478
118. COS Tables and Routing in an SNI Back-to-Back Configuration	479
119. Address Translation	480
120. Example of Name Translation	486
121. Non-Native Network Type 2.1 Connections	491
122. Example of Communication Management Configuration	498
123. Communication Management Configuration After Conversion	501
124. VTAM Functioning as a Dependent LU Server	503
125. Switched Major Node for a Dependent LU Server	505
126. DLUS Located in Different APPN Subnetwork Than DLUR or PLU	507
127. PLU Exists in or through a Subarea Network	508
128. SSCP Takeover When Adjacent CP Does Not Support CP Name Change	509
129. SSCP Takeover When Adjacent CP Is Another Composite Network Node	511
130. Disjoint Hosts	514
131. Hosts with Subarea Connection	515
132. VR-Based TG Between Composite Network Nodes	516
133. Multiple Contiguous VR-Based TGs	517
134. VR-Based TGs In a Communications Management Configuration	520
135. Class-of-Service Resolution at Multiple Interchange Nodes	522
136. VTAM Attachment to a LAN - No Meshed Connection Definitions	529
137. VTAM Attachment to a LAN - Meshed Connection Definitions Provide Optimal Route Calculation	530
138. VTAM Attachment to a Connection Network Reduces Required Connection Definitions	531
139. VTAM Attachment to a Connection Network Also Enables Optimal Route Calculation	532
140. Type 2.1 Channel Connection between a Composite Network Node and a Network Node	536
141. APPN Host-to-Host Channel Connection	537
142. Leased Connection between Two Composite Network Nodes	538
143. Two Network Nodes Connected Using XCA over a 3172 Interconnect Controller	539
144. Using ICA over X.25 between APPN Nodes	540
145. Using ICA over Token Ring between APPN Nodes	542

146. Using ICA over an SDLC Connection between APPN Nodes	543
147. APPN Subnetworks through APPN Multiple Network Connectivity Support	547
148. Buffer Pool after Initial Allocation and after One Expansion	587
149. How VTAM Uses Input Buffers	592
150. How VTAM Uses Output Buffers	593
151. Effect of DELAY Time on Coattailing	598
152. General I/O Buffer Format	601
153. Using Multiple I/O Buffers to Transfer Single Message	602
154. Multiple-Buffer Considerations	602
155. Default Normal Logo	646
156. Single Line Logo	646
157. Sample Logon Manager Configuration	656
158. Cryptography in Single-Domain Environment That Uses PCF/CUSP	664
159. Cryptography in Single-Domain Environment That Uses ICSF/MVS	665
160. Cryptography in Multiple-Domain Environment (Both Hosts Use PCF/CUSP)	667
161. Cryptography in Multiple-Domain Environment (Both Hosts Use ICSF/MVS)	668
162. Cryptography in Multiple-Domain Environment (Hosts Use Different Cryptographic Services)	671
163. Sample Network Showing Default Route	685
164. Sample Network Using Default for UPARM1	686
165. Sample Network Using TG Profile on Some Links	687
166. Illustration of the Use of Matching and Native Relationships	698
167. MPTN Gateway	699
168. Mixed Protocol Operation	702

Tables

1. Start Options and Node Type Relationship	3
2. MVS Data Sets Containing Information for VTAM	21
3. MVS Data Sets Containing Information for Both VTAM and NCP	23
4. DASD Storage Requirements for Target Minidisks	55
5. Local Modifiable Assemble Files Provided by IBM	87
6. Attaching NCP to VTAM	139
7. Independent LUs	141
8. Attaching Peripheral Nodes to VTAM	142
9. Attaching Peripheral Nodes to NCP	143
10. Attaching other domains to VTAM	145
11. CMC configurations	146
12. Dynamic Reconfiguration and Dynamic Change	150
13. Network Routing for Subarea Nodes	151
14. Compression Values for Example of Data Compression Yield	167
15. VTAM Optional User Facility Definition	247
16. Definitions for Dynamically Configured Devices	289
17. Dynamic Reconfiguration Operations for Valid Major Nodes	291
18. Rules for Multiple Definition of Resources	298
19. Adjacent Node Subarea Requirements for Single-Domain Environment	308
20. Endpoint Node Subarea Requirements for Single-Domain Environment	308

21.	Example of Model Name Table	318
22.	Example of Associated LU Table	319
23.	How Session Parameters Are Identified	321
24.	Correspondence of Methods to Letters Used in Following Figures . . .	339
25.	Levels of VTAM and NCP Capable of Type 2.1 Casual Connection . .	407
26.	Adjacent Node Subarea Requirements for Multiple-Domain Environment	424
27.	Endpoint Node Subarea Requirements for Multiple-Domain Environment	424
28.	Resources That You Communicate with in Other Networks	489
29.	ISTINCLM APPNCOS Values Based on LOGMODE Operand Values.	557
30.	Default TG Characteristics	559
31.	#CONNECT Class of Service LINEROW Values	559
32.	Record Format for SNA Controller	570
33.	Record Format for SNA Controller	571
34.	Record Format for Channel-to-Channel Adapters	572
35.	Record Format for Channel-to-Channel Adapters	573
36.	Record Format for Multipath Channel Connections	575
37.	DISPLAY STORUSE Pools	580
38.	VTAM Buffer Pools	583
39.	Number of Buffers per Page	590
40.	I/O Buffer Size and Number of Buffers per Page	590
41.	Coding the Device-Specific Hexadecimal Data of PSERVIC	616
42.	VSCS Accounting Record Format	648
43.	Connection Type for Selected VTAM and Partner Node Combinations	691
44.	Accounting Purpose Modules	695
45.	User Replaceable or Modifiable Modules	695
46.	VTAM Message Modules	696

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

The following terms, denoted by an asterisk (*) at their first occurrence in this publication, are trademarks of the IBM corporation in this country or other countries or both:

ACF/VTAM	AIX
Advanced Peer-to-Peer Networking	AnyNet
APPN	AS/400
BookManager	CICS
Common User Access	CUA
ES/3090	ES/9000
ES/9370	ESA/370
ESA/390	ESCON
FFST	FFST/MVS
First Failure Support Technology	FFST/VM
Hardware Configuration Definition	GDDM
IBM	MVS/ESA
MVS/XA	NetView
OS/2	PR/SM
PS/2	RACF
Virtual Machine/Enterprise Systems Architecture	SAA
Virtual Machine/Extended Architecture	System/360

System/370
VM/XA
VSE/ESA

VM/ESA
VMSES/E
VTAM

About This Book

This book is intended to help system programmers in a VTAM * environment to install VTAM, define network resources to VTAM, replace user tables and exit routines, and tune VTAM for a specific environment. It contains guidance information for using VTAM resource definition statements and macroinstructions.

Who Should Use This Book

Before implementing a VTAM network, you should be familiar with the basic concepts of telecommunication, Systems Network Architecture (SNA), and VTAM. You should also be familiar with the information in *Planning for NetView, NCP, and VTAM*.

For an overview of the new functions in VTAM V4R2, refer to the *VTAM Release Guide for MVS/ESA*, the *VTAM Release Guide for VM/ESA*, and the *VTAM Release Guide for VSE/ESA*.

How to Use This Book

VM, VSE Use this book in conjunction with the *VTAM Overview for VM/ESA and VSE/ESA* flyer included with this book.

You should use this book to obtain an understanding of the major concepts involved in implementing a VTAM network.

The *VTAM Resource Definition Reference* contains a brief description and shows the exact coding format of each VTAM definition statement, start option, and user table. It serves as a reference book for system programmers who are already familiar with the major concepts involved in implementing a VTAM network. Therefore, after you understand the major concepts presented in this book and you are ready to code your network definitions, refer to the *VTAM Resource Definition Reference*.

Technical terms used in this book are defined in the *VTAM Glossary*.

How This Book Is Organized

“Part 1. Concepts and Installation” contains the following chapters:

- Chapter 1, “VTAM Networking Concepts” describes subarea and Advanced Peer-to-Peer Networking * (APPN *) concepts for VTAM.
- Chapter 2, “Installing VTAM under MVS” describes the steps necessary to install VTAM under the MVS operating system.
- Chapter 3, “Installing VTAM under VM” describes the steps necessary to install VTAM under the VM operating system.
- Chapter 4, “Installing VTAM under VSE” describes the steps necessary to install VTAM under the VSE operating system.

“Part 2. Getting Started with VTAM” contains the following chapters:

- Chapter 5, “Getting Started with VTAM - Overview” presents an overview of VTAM implementation issues common to all VTAM networks.
- Chapter 6, “Getting Started with VTAM - Detail” covers specific implementation issues that may not occur in all networks. Use this chapter as necessary for reference when reading Chapter 5, “Getting Started with VTAM - Overview.”

“Part 3. Subarea Networks and SNA Network Interconnection (SNI)” contains the following chapters:

- Chapter 7, “Implementing a Subarea Network” presents the major concepts involved in implementing a subarea network. This chapter does not repeat the information in Chapters 5 and 6, but rather builds upon that information to describe concepts relevant to a subarea network. Therefore, to implement a subarea network, you should be familiar with the information in Chapter 5 and Chapter 6 (where appropriate) before reading this chapter.
- Chapter 8, “Connecting Multiple Subarea Networks ” presents the major concepts involved in implementing a multiple-network environment using SNA network interconnection (SNI). This chapter does not repeat the information in Chapters 5, 6, and 7, but rather builds upon that information to describe a multiple-network environment. Therefore, to implement a multiple-network environment, you should be familiar with appropriate information in Chapters 5, 6, and 7 before reading this chapter.

“Part 4. Implementing APPN in Your Network” contains the following chapters:

- Chapter 9, “Implementing a Combined APPN and Subarea Network” presents the major concepts involved in implementing a combined subarea and APPN network. This chapter does not repeat the information in Chapters 5, 6, and 7, but rather builds upon that information to describe a combined subarea and APPN network. Therefore, to implement a combined subarea and APPN network, you should be familiar with appropriate information in Chapters 5, 6, and 7 before reading this chapter. Chapter 10 contains further information on implementing APPN in your network.
- Chapter 10, “Implementing an APPN Network” presents the major concepts involved in implementing APPN throughout your network. This chapter does not repeat the information in Chapters 5 and 6, but rather builds upon that information to describe an APPN network. Therefore, to implement an APPN network, you should be familiar with the information in Chapter 5 and Chapter 6 (where appropriate) before reading this chapter.

“Part 5. Tuning VTAM for Your Environment” contains one chapter describing how to tune VTAM for your environment.

This book contains the following appendixes for reference:

- Appendix A, “TSO/VTAM ” describes how to implement the time sharing option for VTAM (TSO/VTAM).
- Appendix B, “VM/SNA Console Support” describes how to implement VM/SNA console support (VSCS).
- Appendix C, “Tuning VM for VTAM” describes how to tune VM for VTAM.
- Appendix D, “Logon Manager ” describes how to implement the logon manager.
- Appendix E, “Cryptographic Keys ” describes how to file cryptographic keys.

- Appendix F, “Command Lists: Dynamic Configuration of Channel-Attached Devices” describes the sample command lists provided for the dynamic definition of channel-attached devices.
- Appendix G, “Message Translation Using the MVS Message Service” describes how VTAM messages can be translated using the MVS Message Service.
- Appendix H, “Forcing an APPN Route in a VTAM Network” describes how to force certain sessions to use a specific path rather than the default path.
- Appendix I, “Border Node Connection Types” lists connection types for various VTAM and partner node combinations.
- Appendix J, “VTAM Restricted Materials” lists modules and macroinstructions that are Restricted Materials of VTAM.
- Appendix K, “VTAM AnyNet Feature” contains an overview of Multiprotocol Transport Networking (MPTN) and the VTAM AnyNet* Feature for V4R2.
- Appendix L, “Communications Manager/2's DLUR Support” contains information on Communications Manager/2's dependent LU requester (DLUR) support.

Examples Used in This Book

Throughout this book are many examples. Be aware that these examples are for illustrative purposes only; ***they are not intended to be used as shown.***

Terms Used in the VTAM Library

Clarification of some terms used in the VTAM library follows:

- The term *Advanced Peer-to-Peer Networking* (APPN) represents VTAM's implementation of APPN.
- The terms *CPCP* and *CP-CP* are similar but have different meanings. *CPCP* is the name of a start option or operand. *CP-CP* refers to CP-CP sessions between control points.
- The term *end node* represents an APPN end node.
- The term *integrated communication adapter* (ICA) is used to represent all of the following, except where it is necessary to use the specific term:
 - Communication adapter
 - Telecommunications Subsystem Controller
 - IBM Token-Ring Subsystem Controller
 - Workstation Subsystem Controller
 - Multi-Protocol Communication Subsystem Controller.
- The term *network node* represents an APPN network node.
- The term *VTAM* refers to the VTAM program for V4R2 unless otherwise stated.

Symbols Used in the VTAM Library

The following symbols are used in the VTAM library to indicate information that applies only to a particular operating system or processor:

MVS	Indicates information that applies only to MVS/ESA*.
VSE	Indicates information that applies only to VSE/ESA*.
VM	Indicates information that applies only to VM/ESA*.

These symbols either precede or follow unique information in the VTAM library. Following is an example of how these symbols are used:

VM VSCS is a VTAM application program and must be defined to VTAM.

Artwork Used in This Book

Figure 1 shows the conventions used in this book to illustrate the parts of a network.

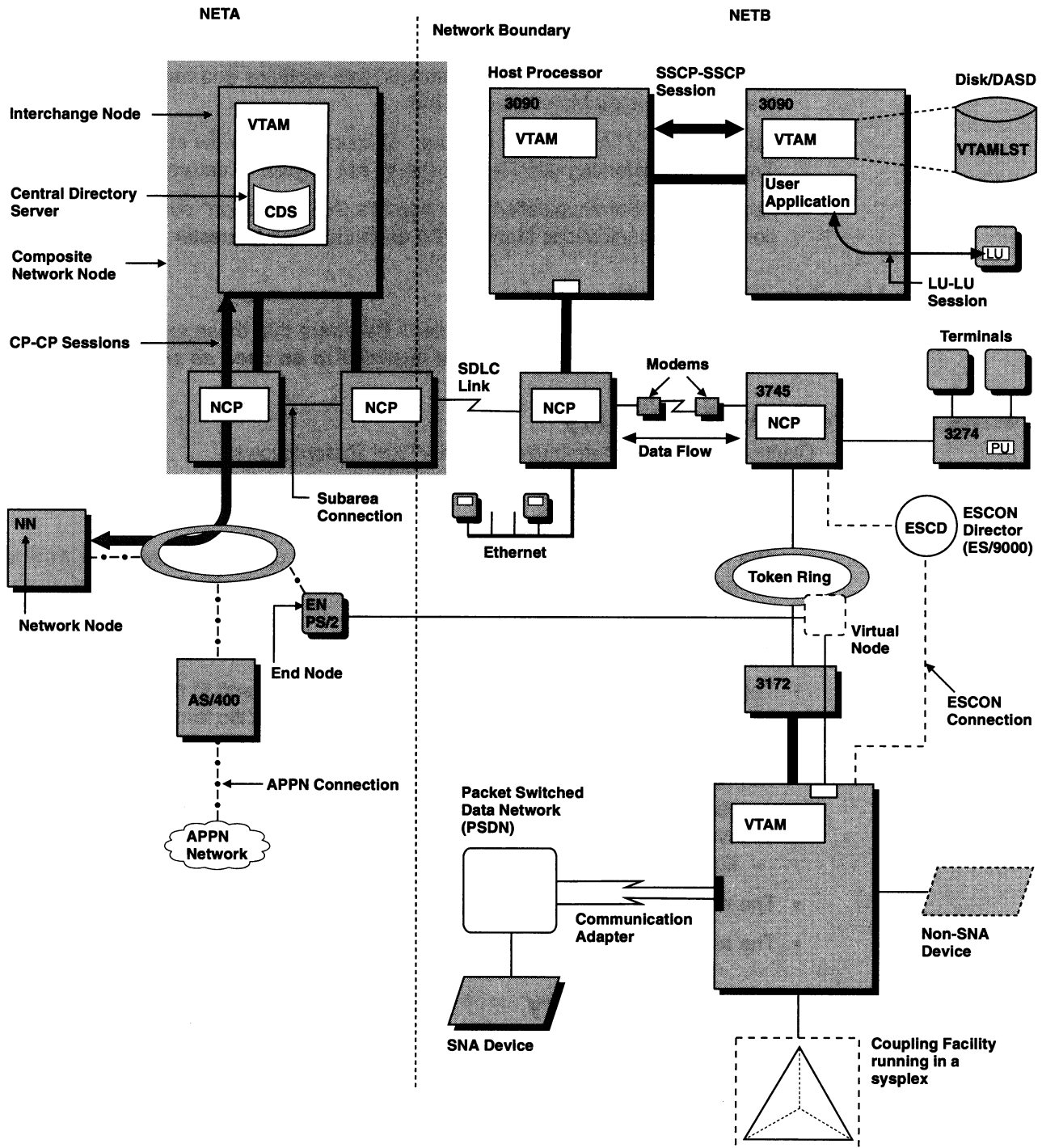


Figure 1. Conventions Used in Network Illustrations

What Is New in This Book

Information has been added to this book to reflect the new functions in VTAM V4R2 for VM and VSE *. Many of the functions were introduced for MVS in V4R1 and V4R2. Changes have also been made as a result of user comments.

The glossary has been deleted from this publication. For definitions of the terms and abbreviations used in this book, refer to the *VTAM Glossary*.

This book was last published in:

- **MVS** May 1994 with order number SC31-6494-00
- **VM, VSE** March 1992 with order number SC31-6434-1

The following sections contain summaries of the changes that have been made for each new function.

Installation

Installing VTAM using the VMSES/E Tool (VM)

Installing VTAM under VM is now done using VMSES/E *. See Chapter 3, “Installing VTAM under VM” on page 51.

LU 6.2 Application Program Interface (API) Enhancements

LU 6.2 Full-Duplex

Sessions between LU 6.2 session partners can now be full-duplex. For information on LU 6.2 full-duplex, see “LU 6.2 Sessions” on page 196.

Network Dynamics

Expanded Dial Information

For an example of the use of DLCADDR, see “NCP/Token-Ring Interconnection (NTRI)” on page 256.

Advanced Peer-to-Peer Networking (APPN) Support

For a basic overview of both APPN and subarea networks, see Chapter 1, “VTAM Networking Concepts” on page 3.

For information on implementing an APPN network, see Chapter 10, “Implementing an APPN Network” on page 525.

For information on implementing a combined APPN and subarea network, see Chapter 9, “Implementing a Combined APPN and Subarea Network” on page 497.

To add an APPN network to your subarea network, use the information from both Chapter 10, “Implementing an APPN Network” on page 525 and Chapter 9, “Implementing a Combined APPN and Subarea Network” on page 497.

Dynamic Reconfiguration of PUs on a Frame Relay Line

For an overview of this function, see “Dynamic Reconfiguration and Change of Operands” on page 149.

Dynamic Reconfiguration of Retry Values and 3174 Group Addresses on a Nonswitched SDLC Line

For information on this function, see “Adding Resources” on page 296.

Operation Interface Enhancements

Start List Processing Enhancements

For information on how start lists are processed, see “Start Options” on page 127.

User-Defined Message-Flooding Prevention Table

For a general description of message flooding prevention, see “Monitoring the Domain” on page 350.

Performance Enhancements

Substitute Class-of-Service Name

For information on how you can use substitute class-of-service (COS) parameters for a session requesting an unknown COS name, see “How Session Traffic Is Assigned to a Specific Route” on page 414.

Non-Disruptive Deactivation of Cross-Domain Resource Managers

For information on specifying whether to keep active sessions when you deactivate cross-domain resource managers, see “Deactivating Resources” on page 428.

Multipath Channels between Host Processors

For information on using multipath channels, see “Multipath Channel (MPC) Connections” on page 266.

Delayed Activation of Logical Lines

For information on implicitly activating a logical line when the physical resources are explicitly activated, see “Delayed Activation of Logical Lines” on page 375. For information on how the OWNER operand affects delayed activation of logical lines, see “Delayed Activation of Logical Lines” on page 430.

Authorized Transmission Priority for LEN Connections

For general information on using the AUTHLEN start option or the AUTHLEN operand on the GROUP, LINE, or PU definition statements, see “Authorized Transmission Priority for LEN Connections” on page 227.

Automatic Logon Enhancements

For information on automatic logon and automatic logon enhancements, see “Automatic Logons” on page 158 and “Automatic Logons” on page 323.

Data Compression

For information on data compression, see “Data Compression” on page 163.

Data Compression Enhancements (VSE)

For information on data compression enhancements, see “Data Compression” on page 163.

Delayed Disconnection

For information on delayed disconnection, see “Controlling Disconnection” on page 255.

Eliminating and Reducing Searches for Unavailable Resources

For information on eliminating and reducing searches for unavailable resources, see “Improving VTAM Performance Using Start Options” on page 131.

Session Limits for Switched Resources (MAXSESS)

For information on session limits for switched resources (MAXSESS), see “Limiting Sessions” on page 227.

Problem Diagnosis

Capturing of Diagnosis Information with First Failure Support Technology (FFST) (VM)

For information about this function, see “First Failure Support Technology (FFST)” on page 104.

DISPLAY STORUSE Pools

If VTAM is in a storage shortage situation, the output from the DISPLAY STORUSE command and Table 37 on page 580 can now be used to determine where excess storage is being used.

System and Configuration Management

Network-Qualified Names

For information on using network-qualified names, see the following:

- For an overview of this function, see “Network Naming Conventions” on page 474.
- For using network-qualified names with CDRSCs, see “Defining Cross-Domain Resources” on page 458.
- To use network-qualified names with the NetView * program, see “Defining the NetView Program” on page 442.
- To use network-qualified names with independent LUs, see “Alias Name Translation and Independent LUs” on page 219.

Adjacent SSCP Lists for CDRSCs

For information on adjacent SSCP lists for CDRSCs, see “Adjacent SSCPs” on page 398.

APPN Host-to-Host Channel

For information on APPN host-to-host channel, see “Using Type 2.1 Channel Connections between APPN Nodes” on page 535.

APPN Multiple Network Connectivity

For information on APPN multiple network connectivity, see “APPN Multiple Network Connectivity” on page 545.

Connection Network

For information on connection network, see “Connection Network” on page 528.

Dependent LU Server

For information on dependent LU server, see “Dependent LU Server” on page 502.

Expanded Addressing Pool

For information on the expanded addressing pool, see “How VTAM Handles Network and Subarea Addressing” on page 306.

Virtual-Route-Based Transmission Groups

For information on virtual-route-based transmission groups see “Virtual-Route-Based Transmission Groups (VR-Based TGs)” on page 515.

Where to Find Information about VTAM

“Bibliography” on page 727 describes the books in the VTAM V4R2 library, arranged according to related tasks. The bibliography also lists the titles and order numbers of books related to this book or cited by name in this book.

Part 1. Concepts and Installation

Chapter 1. VTAM Networking Concepts	3
VTAM Functions	3
VTAM Nodes	3
Nodes with Subarea Function Only	3
Nodes with APPN Function Only	4
Nodes with Both Subarea and APPN Function	8
Network Accessible Units	10
Physical Unit (PU)	10
Logical Unit (LU)	10
Network Control Sessions	11
SSCP-SSCP Sessions	11
CP-CP Sessions	12
End-User (LU-LU) Sessions	13
How VTAM Locates Resources	14
Locating Resources in a Subarea Network	14
Locating Resources in an APPN Network	14
Route Selection	15
Routing in a Subarea Network	15
Routing in an APPN Network	16
Addressing	18
Subarea Addressing	18
APPN Addressing	19
 Chapter 2. Installing VTAM under MVS	 21
Calculating VTAM Virtual Storage Requirements	21
Preparing Your Operating System	21
Defining Data Sets for VTAM	21
Defining VTAM to MVS	31
Loading VTAM	34
Starting VTAM	34
Installing the VTAM Dump Analysis Enhancements and the VIT Analysis Tool	35
Concatenating the Target Data Sets Used in the Installation	36
Installing the ISPF Tables	38
Invoking the ISPF Dialog Tag Language Utility	39
Compiling the Help Panels, Keylists, and Commands	40
Verifying the Dump Formatter Panels	42
Verifying the Trace Formatter Panels	43
Customizing the IPCS Interface	45
Customizing the ISPF Interface	46
Using the Online Message Facility	48
Installing the VTAM Command Set Library for IBM Command Tree/2	48
 Chapter 3. Installing VTAM under VM	 51
Calculating VTAM Virtual Storage Requirements	51
Deciding Where to Run VSCS	51
Determining Storage Requirements	52
Storage Layout	52
Preparing Your Operating System	55
5654010A Directory Entry	55
VTAM Directory Entry	57

Enabling Directory Changes	58
Checkpointing APPN Data Files	58
Profile Requirements for Automatic Logon (AUTOLOG)	60
Discontiguous Shared Segment (DCSS) and Named Saved Systems	63
Loading VTAM - Overview	64
Installation Instructions	66
Installation of VTAM V4R2 for VM/ESA with VMSES/E (VMFINS)	66
VMSES/E Installation Process Overview	66
Place VTAM into Production	76
Defining Discontiguous Shared Segment (DCSS) and Named Saved Systems	82
Generating Group Control System (GCS)	84
Verifying the Installation	85
Applying Service	86
Installing Tables and Modules in VM	86
Service Instructions	88
Trouble Shooting Installation and Service Problems	94
Communication Manager/2's DLUR Support	96
Installing the VTAM VIT Analysis Tool	96
Accessing the Appropriate Disks	96
Updating and Running the ISPF EXEC	97
Installing the ISPF Trace Tables	98
Invoking the ISPF Dialog Tag Language Utility	99
Compiling the Help Panels, Keylists, and Commands	100
Verifying the Trace Formatter Panels	101
Customizing the ISPF Interface	102
First Failure Support Technology (FFST)	104
Installing the Online Message Facility	104
Installing the VTAM Command Set Library for IBM Command Tree/2	104
 Chapter 4. Installing VTAM under VSE	 107
About This Chapter	107
Calculating VTAM Virtual Storage Requirements	107
Preparing Your Operating System	108
Defining Files for VTAM	108
Defining VTAM to VSE	114
Loading VTAM	117
Starting VTAM	117
Keeping an Application Program Active If VTAM Abends	119

Chapter 1. VTAM Networking Concepts

This chapter describes VTAM concepts in both a subarea and APPN environment. It describes the major components of a VTAM network, how they are connected, and the way these components are used in the network.

VTAM Functions

VTAM performs a number of tasks in a network. For example, VTAM:

- Monitors and controls the activation and connection of resources
- Establishes connections and manages the flow and pacing of sessions
- Provides application programming interfaces (for example, an APPC API for LU 6.2 programming) that allow access to the network by user-written application programs and IBM * -provided subsystems
- **MVS** Provides interactive terminal support for TSO.

This chapter presents an overview of the basic concepts of how VTAM performs these tasks.

VTAM Nodes

VTAM can be configured as one of several types of nodes. The type you choose depends on the level of function required for the node and whether the node needs to communicate with other subarea nodes, other APPN nodes, or both. The two start options that determine VTAM's role are HOSTSA and NODETYPE. Table 1 shows how these start options are defined for the different types of nodes.

Table 1. Start Options and Node Type Relationship

Node Type	NODETYPE	HOSTSA
Subarea Node	not coded	<i>subarea number</i>
Interchange Node	NN	<i>subarea number</i>
Migration Data Host	EN	<i>subarea number</i>
Network Node	NN	not coded
End Node	EN	not coded

Nodes with Subarea Function Only

A VTAM subarea node is an SNA type 5 node that functions in a hierarchical environment. Subarea nodes provide services for and control over peripheral nodes. Peripheral nodes require the services of a VTAM subarea node to communicate with other peripheral nodes and subarea nodes in the network. Peripheral nodes include SNA type 2.1, 2.0 or type 1 nodes and function as distributed processors, cluster controllers, or workstations. Type 2.1 nodes can use LEN or APPN connections to communicate with adjacent type 2.1 nodes; however, when attaching to a VTAM subarea node they are restricted to using LEN.

A VTAM subarea node:

- Requires subarea network routing definitions, such as path, virtual route (VR), and explicit route (ER) definitions, to communicate with other subarea nodes
- Uses SSCP-SSCP, SSCP-PU, SSCP-LU, and LU-LU sessions for network control and end-user data
- Does not support CP-CP sessions
- Has a subarea number defined on the HOSTSA start option
- Does not have the NODETYPE start option defined
- Can activate NCPs and have SSCP-SSCP sessions with other VTAM subarea, interchange, and migration data host nodes.

System Services Control Point (SSCP)

An SSCP provides resource management and other session services for end users in a subarea network.

To control and provide services for its subordinate nodes, an SSCP establishes sessions with the PUs and LUs in those nodes. For example, before an LU-LU session can be established, the SSCP must use a directory of network resources to locate the partner LU. When the partner LU is located, the SSCP establishes an SSCP-LU session with that partner LU.

An SSCP provides the following functions:

- Manages resources in a subarea network in accordance with the commands that network operators issue
- Coordinates the initiation and termination of sessions between LUs in separate nodes within its domain or across domains in cooperation with other SSCPs
- Coordinates the testing and status monitoring of resources within its domain
- Performs recovery when communication fails between network components.

CMCs and Data Hosts

A communications management configuration (CMC) simplifies network management. One VTAM in the network acts as the CMC host and owns all resources except applications, which are predominately owned by other hosts called data hosts.

With this configuration the entire network can be controlled by a network operator at the CMC. A communication management configuration can be used to enhance routing. The CMC handles session establishment and termination. The data hosts then spend more time doing application processing and less time doing session establishment and routing.

Nodes with APPN Function Only

A VTAM APPN node is an SNA type 2.1 node that functions in a peer-to-peer environment. APPN nodes use APPN or LEN connections to communicate with other nodes in an APPN network.

A VTAM APPN node:

- Does not require subarea network routing definitions

- Uses CP-CP sessions for network control data and LU-LU sessions for end user data
- Does not support SSCP-SSCP sessions
- Has the NODETYPE start option defined
- Does not have a subarea number defined on the HOSTSA start option
- Cannot activate an NCP
- Requires static definitions only for those resources located within the node
- Requires static definitions only for local connections, not for connections between other nodes in the network.

Control Point (CP)

Each APPN node has a control point that manages the node and its resources. Control points provide function for the APPN network similar to the function the SSCP provides for the subarea network. The control point can implement end node or network node functions.

The control point:

- Activates and deactivates network resources
- Assists in session initiation and termination
- Routes cross-domain requests
- Maintains a dynamic representation of the network in the topology and directory services databases (network nodes only).

In an end node, the CP communicates only with the CP in the network node that provides it with network services. In a network node, the CP communicates with the CPs in adjacent network nodes and with the CPs in adjacent end nodes for which it is the network node server. The control point directs the activation and deactivation of resources and links and helps LUs start and stop sessions.

The control point name of an adjacent APPN node can be statically defined or determined during XID exchange. The control point name of a VTAM node is determined by the SSCPNAME start option.

Network Node

A network node supports its own end users and the end nodes it serves by providing directory and route selection services. It performs intermediate routing of data for sessions that cross it. The network node performs searches of the network to locate resources and calculates the best session route from the node of the primary LU to the node of the secondary LU, based on user-specified criteria.

A VTAM network node:

- Provides intermediate network routing
- Maintains CP-CP sessions with all adjacent network nodes and with all end nodes for which it provides network services
- Has NODETYPE=NN defined
- Does not have a subarea number defined on the HOSTSA start option

- Participates in searches for network resources
- Cannot activate an NCP.

Network nodes and their interconnections form an intermediate routing network. Network nodes use this intermediate routing network to perform searches for partner LUs and calculate the best available route for a session for their own resources and the resources of end nodes for which they are acting as network node servers.

Network nodes are distinguished from end nodes by the functions they provide. A network node provides network services for its own resources and the resources of the end nodes for which it acts as a network node server. For more information on network node servers, see “Network Node Server” on page 7.

To perform these network services, network nodes maintain a topology database and a directory services database. The directory services database contains information on resources in the network, including the network node's local resources, and is updated dynamically as a result of searches for partner LUs. The topology database includes information on network nodes and TGs and is updated dynamically whenever there is a change to the intermediate routing network.

All directory requests from resources (such as LUs on served end nodes, or the network node's own resources) use the directory services of the network node. Therefore, the network nodes are able to collect and control directory information about the APPN network. Network nodes can originate and participate in APPN broadcast searches.

End Node

An end node is an APPN node that relies on the directory and route selection services of a network node to participate in an APPN network.

A VTAM end node:

- Interacts with a network node to request and receive directory and route selection services
- Supports CP-CP sessions between the end node and its network node server
- Can define a network node server list
- Submits search requests to its network node server and obtains the session route from its server when initiating LU-LU sessions
- Dynamically registers its resources with its network node server so that they can be found during broadcast searches
- Has NODETYPE=EN defined
- Does not have a subarea number defined on the HOSTSA start option
- Cannot activate an NCP.

An end node provides communication services for those resources defined within the node. An end node provides directory and session services for its own LUs only.

For directory and routing services to other nodes, an end node uses the services of its network node server. An end node can have links to several network nodes for

connectivity and routing, but only one of these network nodes can be used as the network node server at any given time. A VTAM end node can register its resources in the directory services database of its network node server. This allows searches from other nodes to locate the resources on this end node since VTAM end nodes are not searched. An end node does not perform intermediate routing of sessions; it can only be the origin or destination of a session.

Both a data host and an end node primarily process applications and do not provide network services for the resources attached to them. If your data host has APPN connections and does not require subarea connections, you can migrate it to an end node. (If your data host requires subarea connections, you can migrate it to a migration data host, which is described in “Migration Data Host” on page 8.)

Network Node Server

A network node server is a network node that provides resource location and route selection services to the LUs it serves. These LUs can be on the network node itself, or on end nodes that have established CP-CP sessions with this network node as their network node server.

To support the LUs on served end nodes, a network node server uses the CP-CP session to provide network information for session setup. Network node servers perform searches for partner LUs and calculate the best route to the partner LU.

Any network node can be a network node server for end nodes that are attached to it. The served end nodes are defined as being in that network node server's domain.

Note: A domain in the subarea network is the set of resources controlled by a particular SSCP. A domain in the APPN network is the set of resources served by a particular network node or end node.

Central Directory Server

A central directory server is a network node that builds and maintains a directory of resources from throughout the network. The central directory server provides a similar function to that of a CMC host in a subarea network. The purpose of a central directory server is to reduce the number of network broadcast searches to no more than one per resource. The central directory server is capable of maintaining its directory even when VTAM is halted and restarted.

VTAM network nodes and end nodes can register their resources with a central directory server, which acts as a focal point for resource location information. When a VTAM end node registers its resources, it can request that resources be registered either to its network node server only, or to both its network node server and the central directory server. Entries in a directory database can be registered, defined, or dynamic.

When a network node receives a search request for a resource for which it does not have location information, the network node first sends a directed search request to a central directory server, if there is one. The central directory server searches for information about the location of the resource in its directory. If it does not find the resource's location, the central directory server searches end nodes in its domain, other central directory servers, and, if necessary, the entire network.

If the resource is still not found, the central directory server notifies the node that originally requested the search that the search is unsuccessful.

Nodes with Both Subarea and APPN Function

Nodes that need to be able to directly connect to both subarea and APPN networks require both subarea and APPN function.

Interchange Node

An interchange node resides on the border of an APPN network and a subarea network. It provides protocol conversion between subarea and APPN networks to enable the integration of the two types of networks. Because an interchange node can convert session requests from one protocol to the other and can provide intermediate routing, it can establish sessions from one type of network to the other.

An interchange node combines the function of a subarea node and a network node. It controls resources and functions as a network node in the APPN network and as an SSCP and a cross-domain resource manager (CDRM) in the subarea network. All of the characteristics described for network nodes and subarea nodes apply to interchange nodes.

An interchange node:

- Uses subarea path definitions to determine routes within the subarea network
- Uses the topology database to determine routes within APPN networks
- Uses both SSCP-SSCP and CP-CP sessions to communicate with other nodes
- Has a subarea number and NODETYPE=NN defined
- Can own and activate NCPs.

The interchange node communicates network control data by using SSCP-SSCP sessions with other subarea nodes and CP-CP sessions with other APPN nodes. To enable it to participate in the subarea network it is defined with a unique subarea number and requires subarea path definition statements. It can be connected to other APPN nodes, LEN nodes, and subarea nodes.

Migration Data Host

A migration data host resides on the border of an APPN and subarea network. A migration data host combines the function of an end node with the function and role of a subarea data host.

A migration data host:

- Uses subarea network routing definitions
- Does not perform intermediate routing in APPN or subarea networks
- Uses CP-CP and SSCP-SSCP sessions to communicate with other nodes
- Has a subarea number defined on the HOSTSA start option
- Has NODETYPE=EN defined
- Cannot activate an NCP.

A migration data host is dedicated to processing application programs and does not control network resources, except for local resources like a data host in a subarea network. It also participates as a cross-domain resource manager (CDRM) in the subarea network. The migration data host also functions as an end node in the APPN network. All of the characteristics described for end nodes apply to migration data hosts.

To enable it to participate in the subarea network, it is defined with a unique subarea number and supports subarea path definition statements.

The migration data host communicates network control data by using SSCP-SSCP sessions with other subarea nodes and a CP-CP session with its network node server. It can be connected to other APPN nodes, LEN nodes, and subarea nodes. A migration data host cannot perform intermediate routing in either a subarea network or an APPN network, although its locally attached LEN nodes and dependent LUs can establish sessions with resources in other APPN or subarea domains.

Composite Network Node

A composite network node is composed of a VTAM and any NCPs that it owns. In an APPN network, it functions as a network node and appears to the APPN network as a single node. A composite network node is defined by coding the HOSTSA start option, the NODETYPE start option as NN, and by activating an NCP from that VTAM.

The composite network node can be attached to other APPN nodes and also to subarea nodes. In a composite network node where all external connections are APPN connections, the HOSTSA value is not used in communication outside the node. If the composite node is connected to other nodes by subarea connections, the HOSTSA value is used in communication with the subarea network.

Note: If the composite network node has APPN connections through its NCP, the NCP needs to be at Version 6 Release 2 or greater.

LEN Node

A LEN node is a type 2.1 node that can function in both a subarea and an APPN environment. VTAM and any NCPs that it owns can present the image of a composite LEN node. It appears to all LEN or APPN nodes it is attached to as a single node.

A LEN node provides peer-to-peer connectivity to subarea nodes, end nodes, or network nodes that are providing a LEN appearance. Unlike end nodes, the LEN node cannot establish CP-CP sessions with a network node. Therefore, a LEN node cannot automatically register its resources with a network node server.

A LEN node also cannot establish SSCP-SSCP sessions with a VTAM subarea node. A LEN node can attach to a subarea network as a peripheral node. In a subarea network, LEN nodes can communicate with other LEN nodes without requiring the services of an SSCP.

In an APPN network, you can predefine the CP name of a network node adjacent to a LEN node, which enables the LEN node to send a BIND to the adjacent node and the adjacent node determines the actual location of the LU, calculates the route to it, and forwards the BIND.

Network Accessible Units

Network accessible units (NAUs) are elements in the network with which sessions can be established and queries can be directed. In subarea networks, these units have both a subarea number and an element address and are also called network addressable units. Although subarea addresses are not used in APPN networks, the term network accessible units is used to refer to these units in both types of networks.

Physical Unit (PU)

Physical units (PUs) exist in type 4, type 5, type 2.0, and type 1 nodes. Combined type 2.0/2.1 nodes, such as a 3174 controller, have both a CP and PU.

A physical unit provides the following functions:

- Receives and acts upon requests from CPs, such as activating and deactivating links to adjacent nodes
- Manages links and link stations, while accounting for the unique aspects of different link types
- Manages network addresses, virtual routes, and explicit routes in type 4 and type 5 nodes.

The PU supports sessions with the SSCP in type 5 nodes.

Logical Unit (LU)

A logical unit acts as the intermediary between the end user and the network. Before two end users can communicate, their LUs must be in session with one another. Communication occurs only between LUs of the same LU type. The LU-LU session manages the flow of data between the end users.

VTAM supports LU types 0, 1, 2, 3, and 6.2. These types support the communication requirements of a variety of end users. Each type supports a specific set of communication requirements. The number of the LU type is not related to the PU or node type.

LU types 0, 1, 2, and 3 support communication between application programs and different kinds of workstations. LU type 6.2 supports communication between two programs located at type 5 subarea nodes, type 2.1 peripheral nodes, or both, and between programs and devices. LU type 6.2 supports multiple concurrent sessions, called parallel sessions.

Dependent LU Support

Dependent LUs are supported by VTAM in both subarea and APPN networks.

A dependent LU requires assistance from a system services control point (SSCP) to activate an LU-LU session; therefore, it requires an SSCP-LU session. Both subarea and APPN VTAM nodes provide SSCP support for dependent LUs in their domains. An APPN node provides SSCP support for those LUs in its domain even though it has no SSCP-SSCP sessions with other nodes.

Dependent LUs can use APPN and subarea connections to access VTAM application programs in other domains. For example, a subarea LU type 2 terminal session can be routed to CICS * on a VTAM end node.

To resources in the APPN network, a dependent LU appears to be located on the node that is providing SSCP services to the LU. VTAM can establish SSCP-LU sessions across APPN links; therefore, a VTAM node can be the owner of a dependent LU that is on an attached non-VTAM APPN node, such as an AS/400, * PS/2, * or 3174 controller, regardless of whether the VTAM node has a CP-CP session with the other node.

How Dependent LUs Are Defined: Dependent LUs are owned by a VTAM node and are defined to VTAM. They are considered to be in the domain of the VTAM node. If a type 2.1 node has dependent LUs attached to it, the type 2.1 node must also be defined to VTAM.

Dependent LUs such as terminals and printers appear as cross-domain resources to other VTAM hosts. If the LUs are registered, and if all nodes in the network are connected by CP-CP sessions, the cross-domain resource definition statements in other hosts are not required.

How Dependent LUs Participate in Sessions: Dependent LUs can start a session when the primary LU is owned by VTAM. Dependent LUs can be the secondary LU in sessions with partner LUs in the APPN or subarea network.

Sessions initiated by secondary LUs (such as LU type 2 displays) to a primary LU that is in a VTAM node in the APPN network are supported. Also supported are APPN LUs that initiate sessions with application programs in the subarea network.

Independent LU Support

Independent LUs do not require the services of an SSCP to initiate an LU-LU session. Therefore, they do not have SSCP-LU sessions. Independent LUs can use the services of their local control point to locate and determine a session route to a desired session partner.

Independent LUs must be defined at the LU's owning CP. At the owning CP, they are defined as applications or device-type LUs. Independent LUs can be statically defined at non-owning CPs as cross-domain resources, but this is unnecessary because these resources can be defined dynamically as they are needed. The one exception is when the LU is owned by a LEN node. Because a LEN node cannot register its LUs, these LUs cannot be found as destination LUs if they are not defined to another CP as CDRSCs.

Network Control Sessions

Network control sessions allow two nodes in a network to communicate, enabling end-user sessions between logical units located at those nodes. In subarea networks, network control sessions are called SSCP-SSCP sessions. In APPN networks, network control sessions are called CP-CP sessions.

SSCP-SSCP Sessions

SSCP-SSCP sessions provide control and coordinate session setup for cross-domain and cross-network sessions. These sessions are established between type 5 nodes only. SSCPs exchange capabilities during CDRM activation. SSCP-SSCP sessions are established between all hosts in a network. SSCP-SSCP sessions can only be established between two hosts with subarea function. Within one

subarea network, for an LU-LU session to be established, the SSCPs that own the LUs must be in session with one another.

For dependent LUs, the SSCPs must also establish SSCP-PU and SSCP-LU sessions before an LU-LU session can be established. An SSCP establishes sessions with PUs and LUs defined to it during network activation, or, in the case of switched links, during link activation. SSCP-LU sessions need to be established for dependent LUs only. Independent LUs can function without an SSCP-LU session.

CP-CP Sessions

To perform directory services and topology and route selection services, adjacent APPN nodes throughout the APPN network use a set of parallel CP-CP sessions to exchange network information. CP-CP sessions are established between type 2.1 nodes. A network node or composite network node can establish CP-CP sessions with any network node or composite network node to which it has an APPN link that supports CP-CP sessions. It also establishes CP-CP sessions with each of the end nodes that it serves. End nodes can be attached to several network nodes but can establish a CP-CP session with only one network node server at a time.

When two nodes have one or more CP-CP capable links between them, VTAM requires a CP-CP session to be active between the nodes. If the CP-CP session is deactivated, you need to either reactivate the CP-CP session or deactivate all the CP-CP capable links and reactivate them with CPCP=NO.

In an APPN network you do not need meshed connectivity of CP-CP sessions with every node having a CP-CP session with every other node in the network. With APPN it is only necessary for each APPN node to have a CP-CP session into the network. A CP-CP session is not required between the end points of an LU-LU session. Defining CP-CP capable links to every other node in your network causes excessive topology update flows. It is recommended that you define a primary and backup link with CONNTYPE=APPN and CPCP=YES for each node in your network. It is also recommended that you define additional APPN-capable links with CPCP=NO to provide efficient routing for your LU-LU sessions.

CP-CP sessions between end nodes and their network node servers are used for resource registration and locating destination LUs when the end node wishes to set up a session. CP-CP sessions between two network nodes are used to register resources with a central directory server, perform searches for resources, and exchange topology information.

After a connection has been established, identification information is exchanged between the nodes, and CP-CP sessions are started between the control points in the directly attached nodes.

After the CP-CP sessions are established, the two nodes exchange CP capabilities, such as node type. When both nodes are network nodes, they exchange topology database update (TDU) messages. The TDU messages contain identifying information, node and link characteristics, and resource sequence numbers that identify the most recent updates for each of the resources described in the TDU.

When an LU-LU session is requested, information is transported across the CP-CP sessions to request that an LU-LU session be established between the two session partners. The control points that own the two endpoint resources do not have to

have a direct CP-CP session between them in order to establish the LU-LU session.

End-User (LU-LU) Sessions

End-user sessions are called LU-LU sessions in both subarea and APPN networks. An LU-LU session between two logical units enables one end user to communicate with another. An example of an LU-LU session is an application, such as CICS, communicating with a workstation. LU-LU sessions are used only for user data, not for network control data.

A VTAM node can both initiate sessions and respond to session initiation requests. An LU-LU session is established when a BIND is sent from the primary LU to the secondary LU. Dependent LUs act as secondary LUs only and have an LU-LU session limit of one. Independent LUs can have parallel sessions between the same pair of LUs and can have multiple sessions between one LU and several other LUs. Independent LUs exchange only UNBIND and RSP(UNBIND) for session termination. Independent LUs can be either the PLU or the SLU for any given session. An UNBIND can be sent by either LU.

A VTAM node can have both an SSCP and a CP. Therefore, VTAM nodes support the following types of session initiation over SSCP-SSCP or CP-CP sessions:

- Sessions initiated by the secondary LU
- Queueing of session initiation requests
- Sessions initiated by a third party (CLSDST PASS)
- Orderly termination of existing sessions
- Automatic logon sessions
- Forwarding of session release requests.

In an APPN network, the owning CP for an LU uses CP-CP sessions to obtain location and routing information needed to establish an LU-LU session. The owning CPs do not need a direct CP-CP session with each other if they are not adjacent.

Sessions are established using a specific mode. The mode identifies session parameters and the class of service (COS) to use for the session. In subarea networks, the class of service indicates the virtual route and transmission priority to be used for this session. In APPN networks, the class of service specifies the characteristics required for the session route. These characteristics include the acceptable level of security, cost per byte, cost per connect-time, propagation delay, and effective capacity. For more information about class-of-service in APPN, see “APPN Class of Service” on page 17.

VTAM allows dependent LUs, such as workstations and printers supporting LU types 0, 1, 2, and 3, in one VTAM domain to access VTAM application programs in a different domain by using APPN protocols between domains. All dependent LU types supported by VTAM V3 continue to be supported through APPN networks with one exception: sessions with binary synchronous 3270 devices are not supported over APPN links.

Examples of these dependent LU types are provided in the following list:

LU type 0 3650 and 4700 financial terminals

LU type 1	3767 and 3770 remote job entry (RJE) stations
LU type 2	3270 interactive displays
LU type 3	3270 printers.

How VTAM Locates Resources

For VTAM to locate a resource in the network, VTAM must either have a definition for the resource or be able to dynamically define the resource. Following are descriptions of how resources are located in a subarea or APPN network.

Locating Resources in a Subarea Network

In a subarea network, when VTAM receives a session initiation or directory search request for a resource that is not located in that VTAM's domain, it attempts to locate the resource by sending the request to its adjacent SSCPs.

If the resource is statically defined as a CDRSC and has the CDRM operand defined, VTAM routes the request to the SSCP specified in the CDRM operand. VTAM uses an adjacent SSCP table if:

- The CDRM operand is not defined
- The resource is not found at the CDRM specified
- VTAM has no SSCP-SSCP session with the specified CDRM
- The CDRSC is dynamically defined.

The adjacent SSCP table can be statically defined or, if the start option DYNASSCP is defined as YES (which is the default), it can be dynamic. VTAM searches each SSCP in the table for the resource until the resource is found or the table is exhausted.

In the case of independent LUs located on or through adjacent LEN nodes, VTAM uses these definitions to determine connectivity to the resource when establishing a session with the independent LU.

Locating Resources in an APPN Network

In an APPN network, when VTAM receives a session initiation or directory search request for a resource, VTAM checks its directory services database for the location of the resource. The directory services database contains information on the location of resources obtained through resource definition, resource registration, and as the result of successful searches. If the resource is not located in that VTAM's database, VTAM uses a broadcast search to all other APPN nodes or a search to a central directory server, if one is available. In addition, if there are interchange nodes in the network, at the end of a broadcast search they are sequentially allowed to search the subareas to which they are attached.

Route Selection

Routing in a Subarea Network

Defining a route in a subarea network involves the following elements:

- Subarea links
- Transmission groups (TGs)
- Explicit routes (ERs)
- Virtual routes (VRs)
- Path definitions
- Logon mode tables
- Subarea class of service.

Subarea links are the different types of connections that can be defined between subareas. For example, links include a channel connection between a VTAM and an NCP or an SDLC connection between two NCPs.

Transmission groups allow links with similar characteristics to be logically associated. Several links between the same two subarea nodes might have the same TG number, or the TG number might represent only one link.

Explicit routes consist of a list of TGs and subarea nodes in one direction between the two end points of the route. For each explicit route in one direction, there must be a corresponding explicit route in the opposite direction that follows the same path.

Virtual routes represent the complete route path. Each virtual route has a specific number and the virtual route definition associates that VR number with the explicit route to be used for a session. The virtual route specifies two one-way explicit routes to create a round-trip path.

Explicit routes and virtual routes are defined within path definition statements. Each path definition statement gives the ultimate destination point of the path. The explicit route gives the transmission group and subarea number for the next hop on the path. The association of the virtual route number with the explicit route within the path definition connects the individual hops together into a complete path from one end point to the other.

Each LU-LU session specifies a logon mode table entry. This entry specifies, among other things, the name of a subarea class of service that contains a list of pairs of virtual route numbers and transmission priorities that can be used to establish the session. The subarea class of service allows you to specify the virtual route that best fits the characteristics needed for that session such as capacity, security, and speed.

For more information on subarea routing, see “Routing in a Subarea Network” on page 414.

Routing in an APPN Network

APPN nodes provide a set of topology and routing functions that are used with the directory functions to find resources and calculate routes for sessions. Links must be defined to the nodes that own them, but routes over other links in the network can be learned dynamically.

Topology and Routing Services

The following concepts describe the topology and routing functions:

- Topology database
 - Local
 - Network
- Class of service
- Route selection.

This section describes how APPN nodes provide the topology and routing functions.

Topology Database

To enable a network node to provide routing functions to and from itself and the end nodes it serves, every network node maintains a network topology database that has complete and current topology information about the network. This topology information consists of the characteristics of all network nodes in the network and of all transmission groups (TGs) between network nodes. The end nodes in the network and the TGs connected to them are not considered network topology information.

This section describes the two kinds of topology databases in an APPN network and explains how each is used. Local topology information and network topology information is maintained at each network node. The local topology information is unique to the node; the network topology information is replicated at all network nodes.

Local Topology Database: A local topology database contains information about a node and about the transmission groups connecting the node to other type 2.1 nodes. Both end nodes and network nodes have local topology databases.

In an end node, the local topology database is used to calculate a route to the network node server and to supply the endpoint TG vectors (vectors that describe the TGs connected to the end node) to the network node server. In a network node, the local topology database includes information about the attached end nodes. The local topology database is not saved when VTAM is halted and restarted.

Network Topology Database: The network topology database is referred to as “the topology database.” The network topology database is kept only in network nodes and contains complete and current information about all network nodes in the network, and about all transmission groups connecting them. It does not contain any information about LUs, end nodes, or LEN nodes.

The network topology database dynamically records changes in network topology and can be checkpointed to a file and reloaded when VTAM is restarted in order to reduce the amount of network traffic when a node is restarted.

To keep the topology database current, network nodes in an APPN network send each other topology database updates (TDUs) over CP-CP sessions whenever a resource (node or link) is activated, deactivated, or its characteristics change. Only the current changes are included in the TDU, not the complete topology. The TDUs contain information about the network nodes and the TGs between the network nodes. Every network node receives the TDUs, so all have the same view of the network.

In an APPN network, several mechanisms are used to ensure that unnecessary or excessive TDUs are not propagated throughout the network. For example, multiple TDUs can be grouped and sent out together if there are updates to multiple local TGs.

The topology database is used by the network node to select routes for sessions that originate at the LUs in it and at the end nodes that it serves.

APPN Class of Service

An APPN class of service (COS) defines the required or requested characteristics of a route for a session. A class of service consists of a set of ranges of acceptable values for the characteristics of links and nodes to be used for a session specifying that particular class of service. APPN classes of service are defined in a VTAMLST definition list.

Unlike class of service for the subarea network, where COS is actually a list of virtual routes that are acceptable for a particular class of service, APPN COS specifies the types of routes that are acceptable for a class of service.

This section describes:

- Class-of-service database
- Default class-of-service definitions.

Class-of-Service Database: The class-of-service database exists on all network nodes and end nodes. It is the same on both end nodes and network nodes, and it is built when VTAM is initialized.

The class-of-service database contains:

- Class-of-service definitions from VTAMLST definition files.

The IBM-supplied class-of-service definitions are activated during VTAM initialization. Class-of-service definitions can also be activated with a VARY command or by specifying the name of the VTAMLST member in a CONFIG list. Entries can be added or modified by activating or reactivating the VTAMLST definitions.

User-defined classes of service can also be included. If you want to define your own classes of service, you can do so. For the operands to specify, refer to “Class of Service (COS)” in the *VTAM Resource Definition Reference*.

- A list of logon mode names and the corresponding APPN class of service.

A set of default mappings for the seven IBM-supplied classes of service is included in the default logon mode table. It is recommended that these be used. For information about defining user-specified modes and corresponding classes of service, see “Logon Mode Table” in the *VTAM Resource Definition Reference*.

- A list of calculated routes.

As a network node calculates routes for particular classes of service, the routes are stored in VTAM's memory. Storing routes allows them to be reused, rather than being recalculated each time a route is requested.

Routes are periodically recalculated to allow for distribution of sessions over equally-weighted paths. Start options control this process; see Chapter 4, "Start Options" in the *VTAM Resource Definition Reference*.

Default Class-of-Service Definitions: IBM provides seven default APPN class-of-service definitions. It is recommended that you use the defaults wherever possible and that the class-of-service definitions be the same throughout the network.

The default APPN class-of-service definitions in VTAM are the same default classes of service provided by other APPN products.

Route Calculation and Selection

VTAM provides automatic, preferred route selection based upon dynamic network characteristics and requested class of service. During session establishment, the network node server of the primary LU refers to the topology database to calculate and select the current best route through the APPN network to the secondary LU for the requested class of service.

A network node calculates routes for sessions that originate at the LUs in it and at the end nodes it serves. When a route is calculated, it is stored and can be reused. The stored routes are unique to the node and are never broadcast. The cache of stored routes is maintained automatically. When the maximum cache size (set by the NUMTREES start option) is exceeded, the least recently used routes are discarded first.

Route selection is based on how the actual characteristics of each node and TG along the possible paths match the characteristics required by the requested class of service. The route that a network node selects is the current least-weight (or best) route from the node containing the origin LU to the node containing the destination LU.

Addressing

VTAM uses different addressing schemes in subarea and APPN networks.

Subarea Addressing

Subarea addressing uses an addressing scheme that consists of a subarea number, representing the subarea a resource belongs to, and an element address, which uniquely identifies that resource within the subarea. The subarea number for VTAM is given by the HOSTSA start option. (The subarea number for NCP is determined by the SUBAREA operand on the BUILD statement.) The element address is assigned from a pool of addresses. Both VTAM and NCP maintain a pool of addresses for their attached resources.

APPN Addressing

APPN nodes use local-form session identifiers (LFSIDs) instead of subarea addresses for external communication in an APPN network. VTAM APPN nodes have subarea addresses that are completely internal to the nodes, used by the NCP in a composite node to communicate with its owning VTAM. This subarea address is never used outside the node.

Chapter 2. Installing VTAM under MVS

This chapter describes how to install VTAM under the MVS operating system, which involves:

1. Calculating VTAM Virtual Storage Requirements
2. Preparing Your Operating System
 - a. Defining Data Sets for VTAM
 - b. Defining VTAM to MVS
3. Loading VTAM
4. Starting VTAM
5. Installing the VTAM Dump Analysis Enhancements and the VIT Trace Analysis Tool
6. Using the Online Message Facility
7. Installing the VTAM Command Set Library for IBM Command Tree/2.

Calculating VTAM Virtual Storage Requirements

For information about defining VTAM virtual storage requirements to MVS, refer to the *Estimating Storage for VTAM* diskette.

Preparing Your Operating System

To prepare your MVS operating system to support VTAM, do the following:

- Define data sets for VTAM.
- Define data sets for NCP.
- Define VTAM to MVS.

The logon manager load modules are placed in the SYS1.LINKLIB data set. If you want the load modules to be placed in a separate data set, you need to modify the installation procedure. See "Installing the Logon Manager" on page 654 for more information on adding this data set.

Defining Data Sets for VTAM

This section describes MVS data sets that you need to define for VTAM. Table 2 shows the MVS data sets that contain information for VTAM, and Table 3 on page 23 shows the MVS data sets that contain information for both VTAM and NCP.

Table 2 (Page 1 of 3). MVS Data Sets Containing Information for VTAM

Name of Data Set	Contents	Comments
SYS1.DSDB1	Data files of APPN directory information	Required for APPN directory checkpointing function; must be allocated before VTAM initialization. This data set cannot be allowed to span multiple volumes.

Preparing Your Operating System

Table 2 (Page 2 of 3). MVS Data Sets Containing Information for VTAM

Name of Data Set	Contents	Comments
SYS1.DSDB2	Data files of APPN directory information	Required for APPN directory checkpointing function; must be allocated before VTAM initialization. This data set cannot be allowed to span multiple volumes.
SYS1.DSDBCTRL	Current status of SYS1.DSDB1 and SYS1.DSDB2	Required for APPN directory checkpointing function; must be allocated before VTAM initialization. This data set cannot be allowed to span multiple volumes.
SYS1.DUMPxx	Records of SVC DUMP	Required for diagnosis.
SYS1.LINKLIB	VTAM initialization module, ISTINM01, which is used when VTAM is started	Required; created during VTAM or system installation.
	Logon manager load modules	Required for logon manager
SYS1.LOGREC	VTAM error records	Required; created during VTAM or system installation.
SYS1.LPALIB	VTAM load modules and user-written exit routines to be loaded into the shared link pack area	Required; created during VTAM or system installation.
SYS1.MACLIB	VTAM application program interface macroinstructions	Required; created during VTAM or system installation.
SYS1.NUCLEUS	VTAM resident SVCs and abnormal termination modules	Required; created during VTAM or system installation.
SYS1.PARMLIB	VTAM-related information	Required; created during VTAM or system installation.
SYS1.PROCLIB	JCL for started tasks	Required for logon manager.
SYS1.SISTCLI0	REXX EXECs	Required for VTAM Dump Analysis Enhancements and VIT Analysis.
SYS1.SISTCLS1	VTAM command lists for problem analysis: IPCS CLISTs.	Required; created during VTAM or system installation. See "VTAM Dump Analysis Using IPCS CLISTs" on page 28 and "Diagnosing Problems Using Start Options" on page 133.
SYS1.SISTDAT1	Online tools	Optional; created during VTAM or system installation. Use this library only if you intend to use the online information tools shipped with VTAM. See "Installing the VTAM Command Set Library for IBM Command Tree/2" on page 48.
SYS1.SISTDAT2	Message skeleton file for translation	Required; created during VTAM or system installation. See Appendix G, "Message Translation Using the MVS Message Service" on page 681.
SYS1.SISTDEBUG	Load modules	Required for VTAM Dump Analysis Enhancements and VIT Analysis.
SYS1.SISTMAC1	VTAM macroinstructions used to build user tables and parameter lists to build installation exits	Required; created during VTAM or system installation.

Table 2 (Page 3 of 3). MVS Data Sets Containing Information for VTAM

Name of Data Set	Contents	Comments
SYS1.SISTMSG0	Compiled messages	Required for VTAM DUMP Analysis Enhancements and VIT Analysis.
SYS1.SISTPNL0	Help panels, keylists, and commands	Required for VTAM Dump Analysis Enhancements and VIT Analysis.
SYS1.SISTPNL1	Help panels, keylists, and commands	Required for VTAM Dump Analysis Enhancements and VIT Analysis.
SYS1.TRACE	GTF trace records for VTAM	Required to run VTAM external trace; created during VTAM or system installation.
SYS1.TRSDDB	Network topology database	Required for APPN topology database checkpointing function; must be allocated before VTAM initialization. This data set cannot be allowed to span multiple volumes.
Dynamic I/O configuration data sets	Dynamically created definitions of devices with all associated LUs	Optional; includes USER1.AUTO.VTAMLST and a catalog entry checkpoint data set. Required for dynamic I/O configuration.

Table 3 shows the MVS data sets that contain VTAM information and NCP information if there is an NCP owned by that VTAM.

Table 3 (Page 1 of 2). MVS Data Sets Containing Information for Both VTAM and NCP

Name of Data Set	Contents	Comments
Supply DSNAME of cryptographic data set	VTAM encryption keys	Required for the cryptographic function. (Refer to Appendix E, "Cryptographic Keys " on page 663.)
SYS1.ASAMPLIB	Sample of network operator command table and sample JCL for installation.	Required for installation. Provided by IBM.
SYS1.SAMPLIB	Alterable copy of sample network operator command table, sample JCL for installation, and command lists for dynamic I/O	Required for installation. Provided by IBM.
SYS1.SSPLIB	NCP loader utility program	Required; added when NCP is installed. Refer to "SYS1.SSPLIB" on page 30 for information on SYS1.SSPLIB requirements.
	NCP dump utility program	Required; added when NCP is installed. Refer to "SYS1.SSPLIB" on page 30 for information on SYS1.SSPLIB requirements.
	NCP dump bootstrap program	Required; added when NCP is installed. Refer to "SYS1.SSPLIB" on page 30 for information on SYS1.SSPLIB requirements.
SYS1.VTAMLIB	1. Load modules for VTAM 2. User-defined tables, default tables, and exit routines	Only VTAM load modules are required. Created during system generation. Must be listed in an IEAAPFxx member of SYS1.PARMLIB.

Table 3 (Page 2 of 2). MVS Data Sets Containing Information for Both VTAM and NCP

Name of Data Set	Contents	Comments
SYS1.VTAMLST	VTAM definition statements and start options	Required; created by user before starting VTAM. Use care when modifying this data set (see note on page 25).
Configuration restart data sets	VTAM status of minor nodes for each major node	Required if a warm restart is to be used. Created by user before starting VTAM.
SYS1.NODELST	VTAM status of major nodes	Required if restart of all previously active major nodes is desired.
NCP load library	NCP load modules	Each NCP stored as a separate member of library. Created during NCP generation. Must be an APF-authorized library.
NCP dump data set	Dump records for NCP	Required if VTAM is requested to provide a dump of NCP. Created by user before starting VTAM.
SYS1.LDRIOTAB	Dump records for loader channel I/O trace	Required to hold loader channel I/O trace dumps. Created by user before starting VTAM.
CSP and MOSS dump data set	Dump records for CSP and MOSS	Required if VTAM is requested to provide a dump of CSP or MOSS and if the user wants to store the CSP or MOSS dump in a unique data set. Created by user before starting VTAM.

Key Data Sets Containing Information for VTAM

This section describes the key data sets that contain information for VTAM.

SYS1.VTAMLST: SYS1.VTAMLST is the VTAM definition library, which consists of files containing the definitions for network resources and start options. It is a required partitioned data set, and you need to allocate it on a direct-access volume before you file VTAM network definitions.

This data set can be allocated and cataloged at either of the following times:

- Any time before its initial use. Run the IEHPROGM utility program or the IEBUPDTE utility program.
- When the data set is first used. Code the appropriate job control language (JCL).

To prepare the SYS1.VTAMLST data set, do the following:

1. Allocate space to accommodate the filing of definitions for major nodes and anticipated sets of start options. The amount needed depends on the number of nodes and operands used and on the number of start options. For more information about start options, see “Start Options” on page 127.
2. Specify the DD name for SYS1.VTAMLST as VTAMLST. You should specify the following DCB subparameters:
RECFM=FB,LRECL=80,BLKSIZE=any multiple of 80
3. Code LABEL=RETPD=0 on all DD statements for SYS1.VTAMLST. If you do not, an operator awareness message requiring a reply might be generated.
4. If you generate a NEWDEFN data set as part of NCP generation processing, ensure that it is loaded into SYS1.VTAMLST prior to activating the NCP.

Failure to do so can cause serious problems. VTAM uses the NCP source, in addition to the NCP load module and RRT, when loading and activating communication controllers. SYS1.VTAMLST must contain either the source used as input to the NCP generation process, if a NEWDEFN data set was not created, or the NEWDEFN data set, if one was created. For more information about NEWDEFN, see the *NCP, SSP, and EP Generation and Loading Guide*.

5. If you are configuring VTAM as an APPN node (or plan to do so in the future), copy the COSAPPN and IBMTGPS files from ASAMPLIB into SYS1.VTAMLST. COSAPPN contains the default APPN classes of service and is required if VTAM is configured as an APPN node. IBMTGPS contains the default TG profiles. IBMTGPS is not required, but we strongly recommend that you include it.

Note: Because CP-CP session paths may include subarea VRs, we also strongly recommend that you update your logon mode tables (including the IBM-supplied logon mode table, ISTINCLM) to include an appropriate COS= value on the CPSVCMG and CPSVRMGR mode table entries. Otherwise, a blank COS name will be used to determine the subarea VR and transmission priority that will be used for the VR portion of the CP-CP session path.

Note: You can modify SYS1.VTAMLST, but you need to be very careful about the relationship between VTAM and NCP definition statements. For example, changing a VTAMLST member without changing a corresponding NCP definition statement can cause serious errors that are difficult to diagnose.

SYS1.VTAMLIB: SYS1.VTAMLIB is the VTAM load module library, which consists of files containing the user tables, exit routines, and replaceable constants. It is a required partitioned data set.

To prepare the SYS1.VTAMLIB data set, do the following:

1. Allocate it using the IEHPROGM utility program and catalog it before SMP installation.
2. Define it on a direct-access volume (which can be the system residence volume), and secondary space can be allocated. Space requirements are described in the *VTAM Program Directory*, which is shipped with the VTAM distribution tape.

Use this library to store the following user tables:

- Class-of-service (COS) table
 - Communication network management (CNM) routing table
 - Interpret table containing logon descriptions and any installation-coded logon routines in this table
 - Logon mode table
 - Session awareness (SAW) data filter table
 - Unformatted system services (USS) table.
3. Code the DD name for SYS1.VTAMLIB as VTAMLIB. You should specify the following subparameters on the DCB parameter, with BLKSIZE specified as full-track blocking relative to the capacity of your direct access storage device (DASD):

RECFM=U, BLKSIZE=

4. Define SYS1.VTAMLIB as an authorized library (a library listed in the currently used IEAAPFxx).

Checkpointing APPN Data Sets: The following data sets are used when VTAM is defined as a network node or interchange node and are required for the APPN checkpointing function. These data sets cannot be allowed to span multiple volumes.

- SYS1.DSDB1
- SYS1.DSDB2
- SYS1.TRSDB
- SYS1.DSDBCTRL

SYS1.DSDB1 and **SYS1.DSDB2** contain APPN directory information. This information is used to initialize the directory database when VTAM is restarted.

Directory database information is stored alternately between SYS1.DSDB1 and SYS1.DSDB2. The directory database information is written to one of the data sets whenever a MODIFY CHKPT TYPE=ALL or TYPE=DIR, HALT, or HALT QUICK command is issued.

Not all of the resources from the directory database are written to the data sets when there is a checkpoint. The resources that are written to the data sets are those that:

- Have been the target of a search
- Have a dynamic entry type that is not registered
- Have been updated within a period of time specified by the DIRTIME start option.

The resources that are registered to the database at start up through resource registration and definition are not included in the checkpointed information.

SYS1.DSDBCTRL contains the current status of SYS1.DSDB1 and SYS1.DSDB2. It is read by VTAM during initialization to determine whether SYS1.DSDB1 or SYS1.DSDB2 will be used to load the APPN directory database.

SYS1.TRSDB is required for checkpointing the network topology database. The information in this data set is used to initialize the network topology database whenever VTAM is restarted. The network topology database is written to this file whenever a MODIFY CHKPT TYPE=TOPO or TYPE=ALL, HALT, or HALT QUICK command is issued.

These data sets should be allocated and cataloged prior to VTAM initialization.

To prepare the APPN checkpointing data sets, do the following:

- Specify the DD name for SYS1.DSDB1 as DSDB1, for SYS1.DSDB2 as DSDB2, for SYS1.DSDBCTRL as DSDBCTRL, and SYS1.TRSDB as TRSDB.
- Specify the following DCB subparameters for SYS1.DSDB1, SYS1.DSDB2, and SYS1.TRSDB:
RECFM=FB,LRECL=1000,BLKSIZE=any multiple of 1000
- Specify the following DCB subparameters for SYS1.DSDBCTRL:
RECFM=FB,LRECL=20,BLKSIZE=20

Note: We recommend that you not modify any of these data sets. We also recommend that you not copy the directory data sets and attempt to use them during initialization of a different node. VTAM fails the initial load of the network topology database if another node's checkpointed data set is used or if the SSCPNAME operand is changed between the two IPLs. If the initial load fails, VTAM can acquire the information dynamically through TDUs.

Configuration Restart: If you want to use VTAM's configuration restart facility, define configuration restart Virtual Storage Access Method (VSAM) data sets. For a description of the configuration restart support, see "Configuration Restart" on page 342.

To set up data sets for the major nodes that you will be using with configuration restart, do the following:

1. Use a DD statement to define a configuration restart VSAM data set for each major node. The *ddname* must match the *ddname* on the CONFGDS operand of either the PCCU definition statement for the associated NCP or the VBUILD definition statement for the associated major node. There are no VTAM restrictions on this data set name.

The following example defines a catalog entry to allocate space for a VSAM data set to contain the configuration restart data:

```
DEFINE
  CLUSTER(NAME(RESTART) -
    VOL(PUBLIC) -
    KEYS(18 0) -
    DATA(NAME(RESTART.DATA) -
    RECORDS(200 20) -
    RECORDSIZE(46 158)) -
  INDEX(NAME(RESTARTI.INDEX) -
    TRACKS(1))
```

2. Code the INDEX operand on the DEFINE command, or let it default. (See the sample DEFINE command in the above example.) The data set must be indexed.
3. Code KEYS (18 0). A key length of 18 bytes and an offset of 0 bytes are required.
4. Code RECORDSIZE (46 158). The average record size must be 46 bytes, and the maximum record size must be 158 bytes.
5. Make sure that the number of records in the file is equal to the number of minor nodes defined in the major node. When you choose the number of records for a switched major node, include each PATH definition statement. Therefore, the primary allocation should be the number of minor nodes in the major node, and the secondary allocation should be about 0.1 times the number of minor nodes.
6. When you change a major node definition in SYS1.VTAMLST, do not use the WARM start option when activating the new definition for the first time.

Data Sets for Dynamic I/O Configuration: You can dynamically configure channel-attached devices in your network. For a full description of this support, see "Dynamic Configuration of Channel-Attached Devices" on page 147.

To prepare your system to support dynamic configuration of channel-attached devices, complete the following steps during your installation:

1. Define the USER1.AUTO.VTAMLST library as a partitioned data set. You can customize the name of the data set by altering its name in the ISTDEFIN command list. A sample of ISTDEFIN is found in SYS1.SAMPLIB.
2. Concatenate the USER1.AUTO.VTAMLST data set to the SYS1.VTAMLST data set as defined on the VTAMLST DD statement in the VTAM start procedure. You also need to code the AUTO.VTAMLST data set as shared (DISP=SHR).

```
⋮  
//VTAMLST DD DSN=SYS1.VTAMLST,DISP=SHR  
           DD DSN=USER1.AUTO.VTAMLST,DISP=SHR  
⋮
```

USER1.AUTO.VTAMLST is used by ISTDEFIN for storing automatically generated major nodes. Each member of USER1.AUTO.VTAMLST representing a data host will then contain the definition for just one device. A local SNA major node will also include any of its associated LUs.

3. Set the data set control block (DCB) information for this data set with the same values as for the other VTAMLST data sets.
4. Define a catalog entry checkpoint data set (AUTOCKPT) for dynamic configuration support:

```
DEFINE  
  CLUSTER(NAME('VSAM.AUTOCKPT') -  
    VOL(PUBLIC) -  
    KEYS(4 0) -  
    DATA(NAME('VSAM.AUTOCKPT.DATA') -  
    RECORDS(200 20) -  
    RECORDSIZE(24 136)) -  
  INDEX(NAME(VSAM.AUTOCKPT.INDEX) -  
    TRACKS(1))
```

5. Add this data set using the AUTOCKPT DD statement in the VTAM start procedure:

```
⋮  
//AUTOCKPT DD DSN=VSAM.AUTOCKPT,AMP=AMORG,DISP=OLD  
⋮
```

VTAM Dump Analysis Using IPCS CLISTs: Interactive Problem Control System (IPCS) is a component of MVS used for diagnosing software failures. The VTAM IPCS CLISTs are a set of tools you can use for automating VTAM dump analysis. The IPCS CLISTs reside in the SYS1.SISTCLS1 data set.

For information about using these CLISTs for analyzing VTAM dumps, see “Using IPCS with GTF Trace Option (MVS)” in *VTAM Diagnosis*.

To prepare a data set for the IPCS CLISTs, place the IPCS CLISTs in a cataloged partitioned data set that both TSO and IPCS can access. The data set must be on the MVS system where the dump resides. If you have another CLIST library, you can also add these CLISTs, or you can have a separate library concatenated to existing CLIST libraries.

Note: For TSO to access the data set, it must be allocated to the TSO file SYSPROC. For information on allocating data sets to SYSPROC, see *TSO Extensions Version 2 CLISTS*.

First Failure Support Technology (FFST): First Failure Support Technology * helps you diagnose software problems by capturing information about a potential problem when it occurs.

FFST * is an IBM licensed program product and it is installed using a separate program tape and program directory. See the *FFST/MVS Program Directory* for more information on installing FFST.

NODELST: You can define a NODELST data set to maintain a list of major nodes that are active at one time. If you use the NODELST facility, you need to define VSAM data sets. For more information on how NODELST is used, see "Configuration Restart " on page 342.

To define a NODELST data set:

1. Use the DEFINE command to define a catalog entry and allocate space for an indexed cluster:

```
DEFINE
  CLUSTER(NAME(NODLST1) -
    VOL(PUBLIC) -
    KEYS(2 0) -
    DATA(NAME(NODLST1.DATA) -
    RECORDS(120 20) -
    RECORDSIZE(10 10)) -
  INDEX(NAME(NODLST1I.INDEX) -
    TRACKS(1))
```

2. Code the INDEX operand on the DEFINE command, or let it default. (See the sample DEFINE command in the above example.) The data set must be indexed.
3. Code KEYS (2 0). A key length of 2 bytes and an offset of 0 bytes are required.
4. Code RECORDSIZE (10 10). The average record and the maximum record must each have a length of 10 bytes.
5. Make sure that the number of records in the file is equal to the number of major node and dynamic reconfiguration data set (DRDS) file activations that occur from the time VTAM is started until it is halted. This includes major nodes that are reactivated. The primary allocation should be about 1.2 times the total number of major nodes and DRDS files in the network, and the secondary allocation should be about 0.2 times the total number.
6. You can use defaults for all other data characteristics.

Data Sets Containing Information for NCP

This section describes some of the data sets that contain information for NCP. You might need to define these data sets for your communication controller.

NCP Load Library: The NCP load library contains the NCP and the resource resolution table (RRT) load modules.

To load NCP, create an NCP load module data set to allocate space. Cataloging

the data set is optional. To activate the NCP, the NCP load library must also be available so that the RRT can be accessed. Figure 2 on page 30 shows the correlation between the DD statement for the NCP load module data set and the NCP BUILD definition statement.

DD Statement for NCP Load Module Data Set in VTAM Start Procedure

```
.
.
.
//NCPLOAD DD DSN=SYS1.NCPLOAD, DISP=. . .
.
.
```

NCP Definition Statement

```
BUILD .
.
.
LOADLIB=NCPLOAD,
.
.
QUALIFY=SYS1.
.
.
```

Following two conditions must be met:

- *DD name, lowest level qualifier of data set name, and value of LOADLIB operand must match (in this example, these three are NCPLOAD).*
- *If QUALIFY operand is coded, its value must match highest level qualifier of data set name (in this example, these two are SYS1).*

Figure 2. Correlation between DD Statement and NCP Definition Statement

NCP load module data sets must be in an authorized program facility (APF) library. An NCP load module data set can contain more than one NCP.

SYS1.SSPLIB: SYS1.SSPLIB contains the System Support Program (SSP) utilities used by NCP. SYS1.SSPLIB is a required partitioned data set and is added when NCP is installed. It must be in one of the following:

- SYS1.LINKLIB
- A concatenation of SYS1.LINKLIB (a library listed in the currently used LNKSTxx member in SYS1.PARMLIB)
- A STEPLIB in the start procedure. The STEPLIB must specify an authorized program facility (APF) library.

NCP Dump: The NCP dump data set receives the NCP dump output (one data set for each host VTAM). To dump NCP, you need to allocate space for this data set. You can also catalog this data set. The name of the NCP dump data set is defined when NCP is coded.

This dump data set must accommodate a dump of the entire communication controller storage. The size of communication controller storage depends on the model number.

The DD statement defines the dump data set for the communication controller. The *ddname* must match the *ddname* on the DUMPDS operand of the PCCU definition statement for the associated NCP. VTAM has no restrictions on the data set name.

VTAM dump processing fails if the SSP modules that need to be loaded to process the dump are not accessible to VTAM. Refer to “SYS1.SSPLIB” for information on SYS1.SSPLIB requirements. (For a sample STEPLIB DD statement, refer to the example under “Starting VTAM” on page 34.)

For more information about the NCP dump data set, refer to the *NCP, SSP, and EP Diagnosis Guide*

Loader Channel I/O Trace: The loader channel I/O trace data set (LDRIOTAB) receives communication controller channel information if a load of an NCP fails. The information collected includes channel control words, channel status words, and the first 20 bytes of any data associated with a WRITE, WRITEIPL, or WRITEBRK channel command.

The DD statement defines the trace data set for the SSP load utility. The *ddname* must be LDRIOTAB, but there are no restrictions on the data set name. The data requires only one track of DASD storage and should have a blocksize and logical record length of 121. The data set must be allocated before it is defined in the VTAM start procedure.

Set the disposition as share, pass, and keep in the VTAM start procedure. (For a sample STEPLIB DD statement, refer to the example under “Starting VTAM” on page 34.)

Refer to the *NCP, SSP, and EP Diagnosis Guide* for more information about the loader channel I/O trace data set.

CSP and MOSS Dump (IBM 3720, 3725, and 3745 Only): The CSP and MOSS dump data sets, which apply only to the IBM 3720, 3725, and 3745 Communication Controllers, are used for traces of the communication scanner processor (CSP) and the maintenance and operator subsystem (MOSS). To dump the CSP and MOSS microcode for problem determination, create one data set for the dump of each component. These data sets can be cataloged. The names of these data sets are defined to VTAM in the start procedure. “Starting VTAM” on page 34 contains an example of a start procedure.

The DD statement for each dump data set defines it for the NCP utility used to dump the communication controller. The *ddname* must match the *ddname* on the CDUMPDS (for a CSP dump) or MDUMPDS (for a MOSS dump) operand of the PCCU definition statement for the appropriate NCP. VTAM has no restrictions on the data set name.

Defining VTAM to MVS

Before you can use VTAM, it must be identified to MVS during system generation. After you identify VTAM to MVS, you do not have to perform an MVS system generation when you are migrating to a new release of VTAM. For information on planning and generating VTAM support, refer to the *MVS/ESA System Generation Reference*.

If a system generation is required to identify VTAM to MVS, do the following:

1. In the DATASET macroinstruction, name the MVS data sets used by VTAM, NCP, and any SNA terminal subsystem requiring host support. MVS data sets are described in Table 2 on page 21.
2. Code the following in the DATAMGT macroinstruction to include VTAM into MVS:

ACSMETH=VTAM or IND=YES

3. In the SECONSLE macroinstruction, if the system is generated with the multiple console support option, define a message routing code of at least ROUTCDE=(1,2,8) and a command code of VALDCMD=(1,2). Depending on the message, VTAM also uses routing codes 4, 6, and 10 in combination with basic codes 1, 2, and 8.
4. Message routing codes determine the console at which messages will appear. If the routing codes provided for VTAM messages do not meet your needs, you can change the routing codes used on the messages by coding a system user exit routine (if there is multiple console support). The exit routine receives control before messages are routed so it can examine the messages' routing codes (and descriptor codes) and change them. The system uses the modified routing codes to route these messages. To change routing codes, do the following:
 - a. Prepare the write-to-operator/write-to-operator-with-response (WTO/WTOR) exit routine, and add it to the control program. The WTO/WTOR exit routine can be inserted into the resident portion (communications task) of the control program either before or after system generation.
 - b. Refer to "Message Routing Codes" in *VTAM Messages and Codes* for the message routing codes, and decide on the new routing codes that you want to assign to each message.
5. If you are adding channel-attached devices, you can define these devices using the Hardware Configuration Definition * (HCD) to dynamically add the devices. See "Dynamic Configuration of Channel-Attached Devices" on page 147 for general information about dynamically adding channel-attached devices, and see "Data Sets for Dynamic I/O Configuration" on page 27 for installation information. However, if you are using MVS/CP and are adding channel-attached devices either as part of system generation or as an I/O generation, then you need to modify the IODEVICE macroinstruction.

Note: You cannot use MVS/CP and HCD simultaneously to define channel-attached devices. Use only one or the other.

Include the following in the IODEVICE macroinstruction:

- Channel-attached communication controllers and other channel-attached devices. Devices attached to communication controllers are not defined in IODEVICE macroinstructions; they are defined by definition statements filed for use by VTAM and the NCP.
 - If the NCP is generated with partitioned emulation programming (PEP), the PEP lines are defined with the IODEVICE macroinstruction.
 - If you are using a 2-channel switch on communication controller channel adapters, use CUNUMBR in the MVSCP generation and CNTLUNIT in the IOCP generation to establish the primary and secondary controller addresses.
6. Be sure that the extended CSA value, defined by the second value on the CSA parameter in member IEASYSxx, is at least 4 megabytes for VTAM's use plus anything else needed by the system. For more information, see the *Estimating Storage for VTAM* diskette.
 7. In the CTRLPROG macroinstruction, set CSA equal to or greater than 800.
 8. Name SYS1.VTAMLIB in the APFLIB operand of the IODEVICE macroinstruction when doing a system generation.

Following is an example of system generation statements for including VTAM in MVS:

```
//MVSGEN JOB MSGLEVEL=(1,1),TIME=1440
//      EXEC PGM=ASMBLR,REGION=1042K
//SYSLIB DD DSN=SYS1.AGENLIB,DISP=SHR,UNIT=3330,VOL=SER=DLIB01
//SYSUT1 DD UNIT=3330,SPACE=(CYL,(30,10))
//SYSUT2 DD UNIT=3330,SPACE=(CYL,(30,10))
//SYSUT3 DD UNIT=3330,SPACE=(CYL,(20,10))
//SYSPUNCH DD UNIT=3330,VOL=SER=Volser,DISP=SHR,DSN=dsname
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
```

* SYSTEM CHANNELS

CHANNEL macroinstructions go here

* I/O DEVICES BY CHANNEL ADDRESS

IODEVICE macroinstructions go here

* SYSTEM OPTIONS

DMGT DATAMGT ACSMETH=(VTAM),IND=YES

CTLPGM CTRLPROG CSA=800,
 APFLIB=(SYS1.VTAMLIB,volser,SYS1.INDMAC,volser,qual.ncplib,volser)

* DATASETS

VTAMLIB DATASET VTAMLIB,VOL=(volser,devtype),SPACE=(CYL(p,s,d))
 (plus all other required DATASET macroinstructions)

* CONSOLES

MSTRC CONSOLE MCONS=009,ALTCONS=3E1,ROUTCDE=ALL

SECCON CONSOLE SECONS=3E1,ALTCONS=009,ROUTCDE=ALL,VALDCMD=(1,2,3)

* GENERATION MACROINSTRUCTION

GENERATE macroinstruction goes here

END

/*

If IND=YES or ACSMETH=(VTAM) or both are coded on the DATAMGT statement, VTAM is included in the system.

Starting VTAM

The values of p, s, and d on the DATASET macroinstruction, which specify the direct access space requirements for VTAMLIB, represent the following:

- Primary storage
- Secondary storage
- Directory blocks.

For these space requirements, refer to the *VTAM Program Directory*.

Loading VTAM

To load VTAM, you must download the contents of the VTAM tape. The *VTAM Program Directory* that is shipped with VTAM describes the procedure for installing VTAM from the distribution tape, including suggested coding statements. The *VTAM Program Directory* also describes the contents of the tape and space allocations required on system data sets. The tape contains:

- VTAM load modules
- VTAM macroinstructions (including those for TSO/VTAM) needed to make VTAM part of MVS
- Logon manager load modules
- Sample command lists for use with dynamic I/O
- Command database
- VTAM DUMP Analysis Enhancements and VIT Analysis.

Use the System Modification Program (SMP) to install VTAM. For information on SMP, see the *OS/VS System Modification Program (SMP) Programmer's Guide*.

Starting VTAM

You should code a VTAM start procedure and save it in SYS1.PROCLIB. The system operator specifies the procedure when starting VTAM.

The start procedure is called NET. The name NET is not required but is strongly recommended for consistency in entering the VTAM operator commands and to reduce the operator's chances of making a syntax error. The procedure name you specify must be the first operand on the START, MODIFY, DISPLAY, HALT, and VARY operator commands.

Following is an example of job control statements for a typical start procedure.

```
//NET      PROC
//VTAM     EXEC PGM=ISTINM01,TIME=1440,REGION=4096K,
//STEPLIB DD  DSN=SYS1.SSPLIB,DISP=SHR
//FFSTLIB DD  DSN=SYS1.VTAMLIB,DISP=SHR
//VTAMLST DD  DSN=SYS1.VTAMLST,DISP=SHR
//         DD  DSN=USER1.AUTO.VTAMLST,DISP=SHR
//AUTOCKPT DD  DSN=VSAM.AUTOCKPT,AMP=AMORG,DISP=OLD
//VTAMLIB DD  DSN=SYS1.VTAMLIB,DISP=SHR
//NCPDUMP DD  DSN=SYS1.NCPDUMP,DISP=SHR
// * DATA SETS FOR APPN DATABASE CHECKPOINTING
//DSDB1    DD  DSN=SYS1.DSDB1,DISP=SHR
//DSDB2    DD  DSN=SYS1.DSDB2,DISP=SHR
//DSDBCTRL DD  DSN=SYS1.DSDBCTRL,DISP=SHR
```

```
//TRSDDB DD DSN=SYS1.TRSDDB,DISP=SHR
//* DATA SETS 3720, 3725, AND 3745 DUMPS
//LDRIOTAB DD DSN=SYS1.LDRIOTAB,DISP=(SHR,PASS,KEEP)
//CSPDUMP DD DSN=SYS1.CDUMP,DISP=SHR
//MOSSDUMP DD DSN=SYS1.MDUMP,DISP=SHR
//SYSABEND DD SYSOUT=A
//NCPLOAD DD DSN=SYS1.NCPLOAD,DISP=SHR
//* NODELST DATA SET
//NODLST1 DD DSN=VSAM.NODLST1,AMP=AMORG,DISP=OLD
//* ALTERNATE NODELST DATA SET
//NODLST2 DD DSN=VSAM.NODLST2,AMP=AMORG,DISP=OLD
//* RESTART DATA SET
//RESTART DD DSN=SYS1.RESTART,AMP=AMORG,DISP=OLD
```

Note: On the EXEC statement, ISTINM01 is the main VTAM initialization module name. Code PGM=ISTINM01.

This example is based on the following assumptions:

- The node is APPN-capable.
- A communication controller is in the network.
- The generated NCP and RRT modules for the communication controller reside in data set SYS1.NCPLOAD (NCPLOAD DD statement). The NCP source is in SYS1.VTAMLST.
- The SSP modules needed to load and dump the communication controllers are in SYS1.SSPLIB (STEPLIB DD statement).
- A dump data set is needed for the communication controller (NCPDUMP DD statement).
- The dynamic configuration of channel-attached devices facility (dynamic I/O) is being used (VTAMLST DD and AUTOCKPT DD statements).
- A dump data set is needed for the communication scanner processor (CSP) (CSPDUMP DD statement). For 3380 DASD, this data set should be allocated with at least 7 cylinders using a block size of 512.
- A dump data set is needed for the maintenance and operator subsystem (MOSS) (MOSSDUMP DD statement). For 3380 DASD, this data set should be allocated with at least 10 cylinders using a block size of 512.
- Two NODELST data sets have been defined and can be used by VTAM (NODEDS1 DD and NODEDS2 DD statement).
- Space has been allocated for all data sets, and they have been cataloged.

Installing the VTAM Dump Analysis Enhancements and the VIT Analysis Tool

The following steps should be completed to install the VTAM dump analysis enhancements and the VIT analysis tool:

1. Concatenate the target data sets into the DDNAME statements.
2. Install the dump tables and trace tables.
3. Compile the help panels, keylists, and commands for both dump and trace.
4. Verify that the dump formatter panels have been set up correctly.
5. Verify that the trace formatter panels have been set up correctly.

If problems are encountered, see “VTAM Internal Trace (VIT) Analysis Tool Problems (MVS) (VM)” and “VTAM Dump Analysis Tool Problems (MVS)” in *VTAM Diagnosis* for further information.

Concatenating the Target Data Sets Used in the Installation

The following target data sets contain the necessary data to set up the VTAM Dump Analysis and the VIT Analysis Tool. You need to concatenate the target data sets into the DDNAME statements shown.

Target Data Set	Action	DDNAME	Comment
SYS1.SISTCLI0	Concatenate	SYSPROC	Contains REXX EXECs
SYS1.SISTDEBUG	Concatenate	STEPLIB	Contains load modules
SYS1.SISTPNL0	Concatenate	ISPLLIB	Contains compiled panels
SYS1.SISTMSG0	Concatenate	ISPMLIB	Contains compiled ISPF messages
user-defined data set	Concatenate	ISPTLIB	This data set can be a new or existing one and needs to be the same data set used for ISPTABL.
user-defined data set (same as used for ISPTLIB)	Replace	ISPTABL	Because ISPTABL can only point to one data set, this data set needs to replace any previous data set (or DDNAME for ISPTLIB).

Notes:

1. The SYS1.SISTPNL0 data set should be allocated with enough space to include the panels from SYS1.SISTPNL1.
2. New tables are placed in the first data set under DDNAME ISPTABL.

Use a LOGON PROC to concatenate the data sets. If you create a new LOGON PROC, you need to log off and back on for the PROC to take effect.

The following figure shows an example of a LOGON PROC:

```
//VTAMPROC EXEC PGM=IKJEFT01,TIME=1440,REGION=2048K,
//      PARM='SPFTPO'
//STEPLIB DD DSN=USER.LINKLIB,DISP=SHR
//      DD DSN=SYS1.ISRLOAD,DISP=SHR
//      DD DSN=SYS1.ISPLOAD,DISP=SHR
//      DD DSN=SYS1.LINKLIB,DISP=SHR
//      DD DSN=SYS1.SISTDBUG,DISP=SHR
//SYSABEND DD SYSOUT=A
//SYSUADS DD DSN=SYS1.UADS,DISP=SHR
//SYSHELP DD DSN=SYS1.HELP,DISP=SHR
//SYSPROC DD DSN=USER.CLIST,DISP=SHR
//      DD DSN=SYS1.ISRCLIB,DISP=SHR
//      DD DSN=SYS1.CLIST,DISP=SHR
//      DD DSN=SYS1.SISTCLI0,DISP=SHR
//      DD DSN=SYS1.ISPEXEC,DISP=SHR
//ISPLLIB DD DSN=SYS1.SGIMLMD0,DISP=SHR
//HACPKPT DD DSN=SYS1.HASPKPT,DISP=SHR
//ISPPLIB DD DSN=USER.PANELS,DISP=SHR
//      DD DSN=ISF.V1R1M0.ISFPLIB,DISP=SHR
//      DD DSN=SYS1.ISRPLIB,DISP=SHR
//      DD DSN=SYS1.ISPPLIB,DISP=SHR
//      DD DSN=SYS1.SBLSPNL0,DISP=SHR
//      DD DSN=SYS1.SISTPNL0,DISP=SHR
//ISPMLIB DD DSN=USER.MESSAGES,DISP=SHR
//      DD DSN=SYS1.ISRMLIB,DISP=SHR
//      DD DSN=SYS1.ISPMLIB,DISP=SHR
//      DD DSN=SYS1.SBLMSG0,DISP=SHR
//      DD DSN=SYS1.SISTMSG0,DISP=SHR
//ISPSLIB DD DSN=SYS1.ISRSLIB,DISP=SHR
//      DD DSN=SYS1.ISPSLIB,DISP=SHR
//ISPTLIB DD DSN=USER.TABLES,DISP=SHR
//      DD DSN=ISF.V1R1M0.ISFTLIB,DISP=SHR
//      DD DSN=SYS1.ISRTLIB,DISP=SHR
//      DD DSN=SYS1.ISPTLIB,DISP=SHR
//      DD DSN=SYS1.SMPE.TABLES,DISP=SHR
//ISPTABL DD DSN=USER.TABLES,DISP=SHR
//ISPEXEC DD DSN=SYS1.ISPEXEC,DISP=SHR
//SMPTABL DD DSN=SYS1.SMPE.TABLES,DISP=SHR
//ISPCTL1 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//      DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPCTL2 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//      DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPLST1 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//      DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//ISPLST2 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//      DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
```

Figure 3 (Part 1 of 2). Example of a LOGON PROC

```
//SYSPRINT DD TERM=TS
//SYSIN DD TERM=TS
//DD1 DD DYNAM
//DD2 DD DYNAM
//DD3 DD DYNAM
//DD4 DD DYNAM
//DD5 DD DYNAM
//DD6 DD DYNAM
//DD7 DD DYNAM
//DD8 DD DYNAM
//DD9 DD DYNAM
//DD10 DD DYNAM
//DD11 DD DYNAM
//DD12 DD DYNAM
//DD13 DD DYNAM
//DD14 DD DYNAM
//DD15 DD DYNAM
//DD16 DD DYNAM
//DD17 DD DYNAM
//DD18 DD DYNAM
//DD19 DD DYNAM
//DD20 DD DYNAM
//DD21 DD DYNAM
//DD22 DD DYNAM
//DD23 DD DYNAM
//DD24 DD DYNAM
//DD25 DD DYNAM
```

Figure 3 (Part 2 of 2). Example of a LOGON PROC

Notes:

1. Do not include the source data contained in SISTPNL1 in any ISP library, or problems will occur in using ISTDE01 or ISTTE01.
2. ISPTABL must point to either ISPTLIB or the USER.TABLES data set, and is needed only for compiling the help panels, keylists, and commands.

Installing the ISPF Tables

After the installation data sets have been concatenated, you can install the ISPF tables. To do this, select option 7 from the ISPF/PDF PRIMARY OPTION MENU, and then option 1 from the Dialog Test menu to go to the ISPF INVOKE DIALOG FUNCTION/SELECTION PANEL shown in Figure 4 on page 39.

```

----- INVOKE DIALOG FUNCTION/SELECTION PANEL -----

INVOKE SELECTION PANEL:
  PANEL ===>                                OPT ===>

INVOKE COMMAND:
  CMD ===> ISTDTABL

  LANG ===>                                (APL or blank)
  MODE ===>                                (LINE, FSCR, or blank)

INVOKE PROGRAM:
  PGM ===>                                PARM ===>
  MODE ===>                                (LINE, FSCR, or blank)

NEWAPPL ===> NO                                ID ===>
NEWPOOL ===> NO                                PASSLIB ===> NO
COMMAND ===>
F1=HELP   F2=SPLIT  F3=END    F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP     F8=DOWN   F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
  
```

Figure 4. ISPF Panel Used to Run the Table EXECs

To run all the table EXECs for the formatted dump, enter the EXEC name **ISTDTABL** in the INVOKE COMMAND: CMD ===> field and press Enter.

The individual table EXECs for formatted dump are:

- ISTDT020
- ISTDT025
- ISTDT033
- ISTDT037
- ISTDT053.

To run all the table EXECs for the formatted trace, enter the EXEC name **ISTTTABL** in the INVOKE COMMAND: CMD ===> field and press Enter.

The individual table EXECs for formatted trace are:

- ISTTT007
- ISTTT012
- ISTTT017
- ISTTT024.

Each table EXEC should complete with a return code of 0. RC=0 will appear in the upper right corner of the screen.

The output of the table EXECs is placed in the data set pointed to by the DDNAME **ISPTABL** statement, the user-defined **ISPTLIB** data set.

Invoking the ISPF Dialog Tag Language Utility

To complete the panel setup, you must compile all of the help panels, keylists and commands for both dump and trace. The ISPF Dialog Tag Language Utility, **ISPDTLC**, is used for the compile. **ISPDTLC** is a REXX EXEC provided by ISPF that generates panels, help panels, messages, keylists and commands from Dialog Tag Language source files. For more information about the Dialog Tag Language Utility, see the *ISPF Dialog Tag Language and Reference Guide*. You can enter

ISPD TLC from the ISPF Command ==> line or from the TSO COMMAND PROCESSOR panel, shown in Figure 5 on page 40.

Note: This cannot be done from the TSO READY message.

```
----- TSO COMMAND PROCESSOR -----
ISPF COMMAND ==>

ENTER TSO COMMAND, CLIST, OR REXX EXEC BELOW:

==> ISPD TLC


F1=HELP  F2=SPLIT  F3=END    F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP    F8=DOWN   F9=SWAP   F10=LEFT  F11=RIGHT F12=CURSOR
```

Figure 5. ISPF Panel Used to Invoke the Dialog Tag Language Utility

After ISPD TLC has been invoked, the ISPF Dialog Tag Language Conversion Utility panel is displayed.

Compiling the Help Panels, Keylists, and Commands

The following section describes how to compile the help panels, keylists, and commands using the ISPF Dialog Tag Language Conversion Utility.

To compile the help panels, keylist, and commands, enter the following on the "ISPF Dialog Tag Language Conversion Utility" panel:

- The source data set. ISTDHELP contains a compile list of all the help panels, keylists, and commands for the formatted dump. ISTTHELP contains a compile list of all the help panels, keylists, and commands for the formatted trace.
- The output data set for the panels.
- The output data set for the messages.
- The four-character prefix of the dump or trace application to be used as the keylist application ID.
 - For formatted dump, enter ISTD.
 - For formatted trace, enter ISTT.
- Whether old members are replaced with the new compiled output. Enter YES.
- Whether you want the ISPD TLC compiler messages displayed on the screen or written to a file.

If you indicate that you want the messages written to a data set, you can specify a data set name in the OUTPUT LOG data set field. Otherwise, the output is written to the ISPF log. If you choose to have the messages written

to a data set other than the ISPF log, you need to allocate the output data set before running the compile.

Allocate the output data set with the same attributes as the ISPF log. Unless you specify different data sets for the dump and trace compiles, the messages from the second compile will overwrite the messages from the first compile. Writing the messages to the ISPF log will not cause any messages to be overwritten. It is recommended that you have the messages written to a file.

- Whether you want ISPF messages suppressed. It is recommended that you not suppress messages.
- Your national language.

Examples of the completed ISPF V3R2 panels follow.

To compile the help panels, keylists, and commands for formatted dump, enter the information as provided on the screen shown in Figure 6.

Note: The output data set for the keylists and commands is determined by what was specified for ISPTABL and not by what is specified for the output panel data set.

ISPF Dialog Tag Language Conversion Utility

Enter requested information:

Member name ISTDHELP

DTL source data set - 1 . 'SYS1.SISTPNL1'

DTL source data set - 2 . _____

DTL source data set - 3 . _____

DTL source data set - 4 . _____

Output Panel data set . . 'SYS1.SISTPNL0'

Output Message data set . 'SYS1.SISTMSG0'

Output Log data set . . . _____

(If log data set name is blank, messages go to ISPF log)

Keylist Application Id . . ISTD Up to four characters

Replace old members . . . YES No or Yes

ISPD TLC messages to disk . YES No or Yes

Allow DBCS NO No or Yes

Specify KANA _____ No or Yes or blank

Suppress messages (ISPF) . NO No or Yes

National Language ENGLISH

Command ==> _____

Figure 6. Compiling the Help Panels, Keylists, and Commands for Formatted Dump

To compile the help panels, keylists, and commands for formatted trace, enter the information as provided on the screen shown in Figure 7 on page 42. Be sure to change the member name for the source file, the output message data set if you specified your own, and the keylist application ID.

```

                                ISPF Dialog Tag Language Conversion Utility

Enter requested information:
Member name . . . . . ISTTHelp
DTL source data set - 1 . 'SYS1.SISTPNL1'
DTL source data set - 2 . _____
DTL source data set - 3 . _____
DTL source data set - 4 . _____
Output Panel data set . 'SYS1.SISTPNL0'
Output Message data set . 'SYS1.SISTMSG0'
Output Log data set . . . _____
      (If log data set name is blank, messages go to ISPF log)
Keylist Application Id . ISTT Up to four characters
Replace old members . . YES No or Yes
ISPDTL messages to disk . YES No or Yes
Allow DBCS . . . . . NO No or Yes
Specify KANA . . . . . _____ No or Yes or blank
Suppress messages (ISPF) . NO No or Yes
National Language . . . ENGLISH

Command ==> _____

```

Figure 7. Compiling the Help Panels, Keylists, and Commands for Formatted Trace

After compiling the help panels, keylists, and commands, the VTAM dump analysis enhancements and VIT analysis tool are ready for use. No additional steps are necessary. The following is an example of compiling panel ISTDH0A, including the results of the compile ('1 warning(s) and 0 error(s) found').

```

ISPDTL ISPF Dialog Tag Language Conversion Utility
Converting source file "RDWJA62.ISTD.GML(ISTDH0A)"...
ISPC850W: Warning. Line 33 of file "RDWJA62.TEMPD.PANELS(ISTDH0A)".
  "KEYLIST=ISTDK006" on the "HELP" tag is an ISPF extension to the tag
  language and will not be portable to the OS/2 Dialog Manager. The conversion
  utility will process the attribute "KEYLIST=ISTDK006".
1 warning(s) and 0 error(s) found.
***

```

Figure 8. Messages Resulting from Compilation

Note: Each of the help panels provided generates at least one warning message, ISPC850W. This message is normal when compiling panel ISTDH0A.

After the help panels, keylists, commands, and tables have been generated and/or moved to the correct libraries, the process can now be tested.

Verifying the Dump Formatter Panels

To verify that the Dump Formatter panels are set up correctly, return to the ISPF 7.1 function "INVOKE DIALOG FUNCTION/SELECTION PANEL" and enter the REXX EXEC as shown in Figure 9 on page 43.

```

----- INVOKE DIALOG FUNCTION/SELECTION PANEL -----

INVOKE SELECTION PANEL:
  PANEL ===>                                OPT ===>

INVOKE COMMAND:
  CMD ===> ISTDE01

  LANG ===>                                (APL or blank)
  MODE ===>                                (LINE, FSCR, or blank)

INVOKE PROGRAM:
  PGM ===>                                PARM ===>
  MODE ===>                                (LINE, FSCR, or blank)

NEWAPPL ===> YES                                ID ===> ISTD
NEWPOOL ===> NO                                PASSLIB ===> NO
COMMAND ===>
F1=HELP   F2=SPLIT  F3=END   F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP     F8=DOWN   F9=SWAP  F10=LEFT   F11=RIGHT  F12=CURSOR
  
```

Figure 9. Verifying the Dump Formatter Panels

The first ISPF panel you should see is shown in Figure 10.

```

                                VTAMMAP Analysis Menu

Select one of the following. Then press Enter.

— 1. APPC . . - APPLCONV, PARTNRLU, APPLMODE, APPMODAL
   2. APPN . . - APPNBASE, FNDADJCP, FNDANDCB, FND COS, FNDDECB, etc
   3. General. - HOST, VTAM, VTBASIC, VTFNDMOD, VTMODS, VITAL, etc
   4. Queues . - PABSCAN, VTCVTPAB, VTREADYQ
   5. Resource - RDTCHECK, RDTFULL, RDTHIER, RDTSUM, VTNODE
   6. Session. - FINDDSIB, FINDSIB, SES, SIBCHECK
   7. Search . - SRTFIND
   8. Storage. - SPANC, STORAGE, VTBUF, VTRPH
   9. Waits. . - VTWRE
  10. ERs/VRs. - ROUTES, VTVRBLK
  11. CLISTs. - ISTVABND, ISTVDUMP, ISTVMAP, ISTVSAVE, ISTVSLIP

(C) Copyright IBM Corporation 1993. All rights reserved.
Command ===>
F1=Help   F2=Split  F3=Exit  F9=Swap  F12=Cancel
  
```

Figure 10. Main Menu for Selecting Dump Options

Press the PF1 key to verify that the appropriate help panel is displayed.

Verifying the Trace Formatter Panels

To verify that the Trace Formatter panels are set up correctly, return to the ISPF 7.1 function "INVOKE DIALOG FUNCTION/SELECTION PANEL" and enter the REXX EXEC as shown in Figure 11 on page 44.

```

----- INVOKEDIALOG FUNCTION/SELECTION          FUNCTION RC =  0

INVOKEDIALOG SELECTION PANEL:
  PANEL  ===>                                OPT  ===>

INVOKEDIALOG COMMAND:
  CMD    ===> ISTTE01

  LANG   ===>                                (APL or blank)
  MODE   ===>                                (LINE, FSCR, or blank)

INVOKEDIALOG PROGRAM:
  PGM    ===>                                PARM  ===>
  MODE   ===>                                (LINE, FSCR, or blank)

NEWAPPL  ===> YES                                ID    ===> ISTT
NEWPOOL  ===> NO                                PASSLIB ===> NO
COMMAND  ===>
F1=HELP  F2=SPLIT  F3=END   F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP    F8=DOWN   F9=SWAP   F10=LEFT  F11=RIGHT  F12=CURSOR
  
```

Figure 11. Verifying the Trace Formatter Panels

The first ISPF panel you should see is shown in Figure 12.

```

              VTAM Internal Trace Analysis

Select one of the following. Then press Enter.

—  1. Storage Analysis
    2. Request/response unit counting
    3. VIT extraction
    4. Input complete

(C) Copyright IBM Corporation 1992. All rights reserved.
Command ===>
F1=Help  F2=Split  F3=Exit  F9=Swap  F11=Retrieve F12=Cancel
  
```

Figure 12. Main Menu for Selecting Trace Parameters

Press the PF1 key to verify that the appropriate help panel is displayed.

Note: It is recommended that you position the command line at the bottom of the screen using ISPF PARMs option DISPLAY and changing the 'COMMAND LINE PLACEMENT ===> ASIS' to BOTTOM to improve readability.

Customizing the IPCS Interface

If you want a customized interface to be active to select the VTAM dump analysis commands, customize the IPCS panel BLSPPRIM by adding the highlighted lines in Figure 13 to create and activate option 7 on the IPCS Primary Option Menu as shown in Figure 14 on page 46. This modification allows you to access VTAMMAP directly for dump processing. When this option is selected, control is passed to the the ISTDE01 EXEC. This EXEC controls the IPCS panels for the dump formatter.

```
)ATTR
/* ===== */
/* 5685-001 This panel is "Restricted materials of IBM" */
/* (C) Copyright IBM Corporation 1988 */
/* Licensed materials - property of IBM */
/* Refer to copyright instructions, form number G120-2083 */
/* ===== */
¢ TYPE(INPUT) INTENS(HIGH) CAPS(OFF) JUST(LEFT) PAD(NULLS)
@ TYPE(TEXT) COLOR(GREEN) INTENS(LOW)
)BODY
%----- IPCS PRIMARY OPTION MENU -----
%OPTION ==>¢ZCMD

% 0 +DEFAULTS - Specify default dump and options @*****
% 1 +BROWSE - Browse dump data set @* USERID - &ZUSER
% 2 +ANALYSIS - Analyze dump contents @* DATE - &ZDATE
% 3 +SUBMIT - Submit problem analysis job to batch @* JULIAN - &ZJDATE
% 4 +COMMAND - Enter IPCS subcommand or CLIST @* TIME - &ZTIME
% 5 +UTILITY - Perform utility functions @* PREFIX - &ZPREFIX
% 6 +DUMPS - Manage dump inventory @* TERMINAL- &ZTERM
% 7 +VTAM - VTAM dump analysis commands @* PF KEYS - &ZKEYS
% T +TUTORIAL - Learn how to use the IPCS dialog @*** *****
% X +EXIT - Terminate using log and list defaults

+Enter%END+command to terminate IPCS dialog
)INIT
&ZPRIM = YES /* Always a primary option menu */
&ZHTOP = BLSPHelp /* Tutorial table of contents */
.CURSOR = ZCMD
.HELP = BLSPHelp
&ZHINDEX = &Z /* No tutorial index is supplied */
)PROC
&PASSLIB = &Z
IF (&ZDBCS = YES, NO)
&PASSLIB = PASSLIB
&ZSEL = TRANS( TRUNC (&ZCMD, '.')
0, 'PGM(BLSGSCMD) PARM(%BLSCSETD)'
1, 'PGM(BLSLDISP) NEWAPPL(BLSL) &PASSLIB'
2, 'PANEL(BLSPSCRN)'
3, 'PANEL(BLSPBKGD)'
4, 'PANEL(BLSPDSLE)'
5, 'PANEL(BLSPUTIL)' /* @00A*/
6, 'PGM(BLSGDUIN)'
7, 'PGM(BLSGSCMD) PARM(%ISTDE01) NEWAPPL(ISTD)'
t, 'PGM(ISPTUTOR) PARM(BLSPTUTR)'
T, 'PGM(ISPTUTOR) PARM(BLSPTUTR)'
' '
x, 'EXIT'
X, 'EXIT'
*, '?' )
)END
```

Figure 13. Sample IPCS Panel BLSPPRIM Customization

```
-----IPCS PRIMARY OPTION MENU-----
OPTION  ==> _

0  DEFAULTS      - Specify default dump and options
1  BROWSE        - Browse dump data set
2  ANALYSIS      - Analyze dump contents
3  SUBMIT        - Submit problem analysis job to batch
4  COMMAND       - Enter IPCS subcommand or CLIST
5  UTILITY       - Perform utility functions
6  DUMPS         - Manage dump inventory
7  VTAM         - VTAM dump analysis commands
T  TUTORIAL      - Learn how to use the IPCS dialog
X  EXIT          - Terminate using log and list defaults

Enter END command to terminate IPCS dialog
```

Figure 14. Addition of Option 7 on the IPCS Primary Option Menu

Customizing the ISPF Interface

If you want a customized interface to be active to select the VTAM trace analysis commands, customize the ISPF panel ISR@PRIM by adding the highlighted lines shown in Figure 15 on page 47 to create and activate option V on the ISPF/PDF Primary Option Menu as shown in Figure 16 on page 48. When this option is selected, control is passed to the ISTTE01 EXEC. This EXEC controls the ISPF panels for the trace formatter.

```

)ATTR
+ TYPE(TEXT) COLOR(GREEN) INTENS(LOW)
)BODY
%----- SAMPLE ISPF/PDF PRIMARY OPTION MENU -----+
%OPTION ==>_ZCMD
%
%
%                                +USERID - &ZUSER
% 0 +ISPF PARMS - Specify terminal and user parameters +TIME - &ZTIME
% 1 +BROWSE - Display source data or output listings +TERMINAL - &ZTERM
% 2 +EDIT - Create or change source data +PF KEYS - &ZKEYS
% 3 +UTILITIES - Perform utility functions
% 4 +FOREGROUND - Invoke language processors in foreground
% 5 +BATCH - Submit job for language processing
% 6 +COMMAND - Enter TSO command or CLIST
% 7 +DIALOG TEST - Perform dialog testing
% 8 +LM UTILITIES- Perform library administrator utility functions
% 9 +IBM PRODUCTS- Same as option S (SER PRODUCTS)
% 10 +SCLM - Software Configuration and Library Manager
% C +CHANGES - Display summary of changes for this release
% V +VTAM - VTAM trace analysis commands
% T +TUTORIAL - Display information about ISPF/PDF
% S +SER PRODUCTS- Southeast Region product options
% I +SER IC TOOLS- Southeast Region Info-Center and Toolkits
% P +RPM - Regional Problem Management
% X +EXIT - Terminate ISPF using log and list defaults
%
+Enter%END+command to terminate ISPF.
%
)INIT
.HELP = ISR00003
&ZPRIM = YES /* ALWAYS A PRIMARY OPTION MENU */
&ZHTOP = ISR00003 /* TUTORIAL TABLE OF CONTENTS */
&ZHINDEX = ISR91000 /* TUTORIAL INDEX - 1ST PAGE */
&ZSCLMPRJ = &Z
VPUT (ZHTOP,ZHINDEX,ZSCLMPRJ) PROFILE
)PROC
&ZQ = &Z
IF (&ZCMD ^= ' ')
&ZQ = TRUNC(&ZCMD, '.')
IF (&ZQ = ' ')
.MSG = ISRU000
&ZSEL = TRANS( &ZQ
0, 'PANEL(ISPOPTA)'
1, 'PGM(ISRBRO) PARM(ISRBRO01)'
2, 'PGM(ISREDIT) PARM(P, ISREDM01)'
3, 'PANEL(ISRUTIL)'
4, 'PANEL(ISRFPA)'
5, 'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
6, 'PGM(ISRPTC)'
7, 'PGM(ISPYXDR) PARM(ISR) NOCHECK'
8, 'PANEL(ISRLPRIM)'
9, 'PANEL(SERPP000)' /* CHANGED HERE? FROM ISRDIIS */
10, 'PGM(ISRSCLM) NOCHECK'
C, 'PGM(ISPTUTOR) PARM(ISR000005)'
V, 'CMD(%ISTTE01) NEWAPPL(ISTT)'
T, 'PGM(ISPTUTOR) PARM(ISR000000)'
S, 'PANEL(SERPP000)'
I, 'PANEL(SERIC000)'
P, 'CMD(%SRRPM)'
, ' '
, ' '
X, 'EXIT'
*, '?' )
&ZTRAIL = .TRAIL
)END

```

Figure 15. Sample ISPF Panel ISR@PRIM Customization

```
----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION  ===>

0 ISPF PARMs - Specify terminal and user parameters      USERID - USERID
1 BROWSE    - Display source data or output listings    TIME    - 9:29
2 EDIT      - Create or change source data              TERMINAL - 3278
3 UTILITIES - Perform utility functions                 PF KEYS  - 12
4 FOREGROUND - Invoke language processors in foreground
5 BATCH     - Submit job for language processing
6 COMMAND   - Enter TSO command or CLIST
7 DIALOG TEST - Perform dialog testing
8 LM UTILITIES- Perform library administrator utility functions
9 IBM PRODUCTS- Same as option S (SER PRODUCTS)
10 SCLM      - Software Configuration and Library Manager
C CHANGES   - Display summary of changes for this release
V VTAM       - VTAM trace analysis commands
T TUTORIAL   - Display information about ISPF/PDF
S SER PRODUCTS- Southeast Region product options
I SER IC TOOLS- Southeast Region Info-Center and Toolkits
P RPM        - Regional Problem Management
X EXIT       - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

Figure 16. Addition of Option V on the ISPF/PDF Primary Option Menu

Using the Online Message Facility

The Online Message Facility is no longer shipped as a library on the product tape. It is available on the *IBM Networking Softcopy Collection Kit* CD-ROM. Information on planning for, installing, and using the facility is in *Using the Online Message Facility*, which is available on that CD-ROM.

Installing the VTAM Command Set Library for IBM Command Tree/2

To install the VTAM command set library for the IBM Command Tree/2 program on a workstation, follow these steps:

1. Download the VTAM command set library files to your workstation.

Use the OS/2 * RECEIVE command to download the files from the SYS1.SISTDAT1 data set. Enter the following commands on an OS/2 command line (where id is the emulator session from which you are downloading):

```
RECEIVE ISTC420M.RAM id:'SYS1.SISTDAT1(ISTC420M)'  
RECEIVE ISTLDRM2.EXE id:'SYS1.SISTDAT1(ISTLDRM2)'
```

OS/2 downloads the ISTLDRM2 unpacking utility and the packed version of the VTAM command set library and stores them on your workstation. For more information on the RECEIVE command, enter HELP "RECEIVE" on the OS/2 command line.

Note: You must be in TSO mode to download these files.

For example, if you are downloading from your A emulator session, and you want to store the VTAM command set library for IBM Command Tree/2 on your D: drive in directory IBMFKB\RUNTIME\ISTC420M, enter:

```
RECEIVE D:\IBMFKB\RUNTIME\ISTC420M\ISTC420M.RAM A:'SYS1.SISTDAT1(ISTC420M)'
```

IBMFKB is the directory that contains the IBM Command Tree/2 program files. It is suggested (but not required) that you install each command set library in a separate subdirectory of IBMFKB\RUNTIME. For example, you could store the VTAM V4R2 command set library in IBMFKB\RUNTIME\ISTC420M, and a user-written command set library in IBMFKB\RUNTIME\MYCMDS.

2. Unpack the VTAM command set library files.

After the packed command set library is on your workstation, you need to unpack the files before you can build VTAM commands. To unpack the files, go to an OS/2 command line and be sure the prompt is set for the directory to which you downloaded the command set library (for example, D:\IBMFKB\RUNTIME\ISTC420M). Now enter:

```
ISTLDRM2 ISTC420M.RAM /D
```

This unpacks the files and stores them in the current directory. The ISTLDRM2 utility does not erase the packed command set library file. For backup purposes, it is recommended that you keep a copy of ISTC420M.RAM on the workstation or SYS1.SISTDAT1(ISTC420M) on the host.

For more information on the IBM Command Tree/2 program, including how to create user profiles to start the program with the VTAM command set, see *Using IBM Command Tree/2*. This book is also available on the *IBM Networking Softcopy Collection Kit* CD-ROM.

Chapter 3. Installing VTAM under VM

This chapter describes how to install and service VTAM under the VM operating system. The product number for VTAM V4R2 for VM/ESA is 5654-010.

If at any time during installation or service you receive a VTAM error message, consult *VTAM Messages and Codes*. If you receive a VMSES/E error message, consult *VM/ESA Messages and Codes*. If you want to tune VM to improve performance while running VTAM, see Appendix C, "Tuning VM for VTAM" on page 649.

Because the installation and service procedures do not rely on the VTAM user ID being present, there is no automatic force of the VTAM user ID during installation or service processing. You must ensure that the VTAM being updated is not in use when you reinstall VTAM or apply service.

Calculating VTAM Virtual Storage Requirements

To calculate VTAM virtual storage requirements, you must first decide if you want VSCS to run in the VTAM virtual machine or in its own virtual machine.

Deciding Where to Run VSCS

You can run VSCS in the VTAM virtual machine or in its own virtual machine. Generally, larger systems should run VSCS in its own virtual machine. On larger systems (for example, with more than 1,000 LUs logged onto VM) you might see real response time benefits by running VSCS in its own virtual machine. However, you will only see these benefits on a system that has adequate real resources to handle the extra requirements.

For more information about VSCS, see Appendix B, "VM/SNA Console Support" on page 627.

Following is some information to help you decide how you want to run VSCS:

VSCS in the VTAM virtual machine:

- Benefits:
 - VSCS can communicate with VTAM more efficiently because cross-machine communication is not necessary.
 - The VTAM operator can issue both VTAM and VSCS commands to the same virtual machine.
 - Generally, the working set size of the virtual machine will be smaller than if there were two virtual machines.
- Drawbacks:
 - Virtual storage shortages might be caused by VSCS sharing the private storage of the VTAM virtual machine.
 - Problem diagnosis of storage problems might be more difficult because messages might not point to the failing component.
 - VSCS might not get sufficient CPU cycles because it must share the virtual processor of that machine with VTAM.

- If the virtual machine must be logged off for one of the components, the other component will also be affected.

VSCS in its own virtual machine:

- Benefits:
 - Problem diagnosis of storage problems might be easier.
 - For systems that have more than one real processor, VSCS and VTAM can run concurrently if necessary because VSCS has its own virtual processor.
 - One virtual machine can be reset without affecting the other virtual machine.
- Drawbacks:
 - Because of cross-machine communication, more overhead is needed to communicate with VTAM.
 - The VTAM operator must be sure to enter VTAM and VSCS commands on the proper virtual machine.
 - A larger VM working set size might be needed, placing additional real storage and DASD requirements on the system.

If you intend to run VSCS in its own virtual machine, then you must calculate the size of the VSCS virtual machine. Follow the same steps used to calculate the size of the VTAM virtual machine. You will also need to create the virtual machine by adding a directory entry, authorizing the virtual machine to GCS, and creating a PROFILE GCS. For information on these steps, see Appendix B, “VM/SNA Console Support” on page 627.

Determining Storage Requirements

To calculate the storage size of the VTAM virtual machine, do the following:

1. Use the *VM/ESA Planning Guide and Reference* or *VM/ESA GCS Reference* to determine the amount of private storage required by GCS. This storage is required for control blocks and work areas that are accessed only by the GCS supervisor.
2. Use the *Estimating Storage for VTAM* diskette to calculate private storage requirements for VTAM and VSCS.
3. Use the *VM/ESA Planning Guide and Reference* or *VM/ESA GCS Reference* to determine the amount of common storage required by GCS. This storage is required for the GCS supervisor code, the GCS trace table, and for a control block for each virtual machine in the group.
4. Use the *Estimating Storage for VTAM* diskette to calculate common storage requirements for VTAM.

Storage Layout

Before updating the VM control program (CP), you must decide on a storage layout for the VTAM virtual machine. You must specify the virtual address range of the VTAM discontinuous shared segment (DCSS) and the virtual address ranges of the GCS shared system.

Because each virtual machine in the GCS group will load both the GCS shared system and the VTAM shared segment, this storage layout will be the same as the storage layout for the other virtual machines in the group. The only difference is the actual size of each individual virtual machine. For example, the storage layout for VSCS will be the same as for VTAM, except for the size of the virtual machines themselves.

The VTAM DCSS contains VTAM modules that are shared among virtual machines in the GCS group. GCS storage consists of two parts: GCS private storage and GCS common storage.

You might not be able to define the VTAM DCSS address range discontinuous from the boundaries of all other virtual machines in the GCS group. Therefore, you can define a storage size that is appropriate for each virtual machine. You will need to consider the following if you find this necessary:

- If the virtual machine size encompasses the VTAM DCSS, the storage occupied by that VTAM DCSS is not available as private storage.
- The storage locations occupied by all DCSSs defined to GCS must not overlap.
- Because the VTAM DCSS is generated while under the control of CMS, VTAM cannot be defined at a location where it will collide with CMS. If it does collide with CMS, CMS will not allow the VTAM DCSS to overlay its storage. Therefore, the VTAM DCSS cannot be defined in the last megabyte of the virtual machine because CMS has some storage allocated there. See “Defining Discontiguous Shared Segment (DCSS) and Named Saved Systems” on page 82 for details on avoiding this problem.

Figure 17 on page 54 shows a sample storage layout with the VTAM shared segment (DCSS) defined as contiguous to the VTAM virtual machine. The size of the virtual machine is 32M, and the amount of private storage is equal to the size of the virtual machine minus storage used for the segments. VTAM private storage is used for VTAM control blocks and buffers. Reducing the size of the VTAM virtual machine will result in less private storage for VTAM. For example, reducing the size of the VTAM virtual machine to 20M reduces private storage by 12M. Reducing the size of the VTAM virtual machine to 16M, however, still reduces private storage by only 12M since GCS High CSA uses the space between 16M and 20M.

Calculating VTAM Virtual Storage

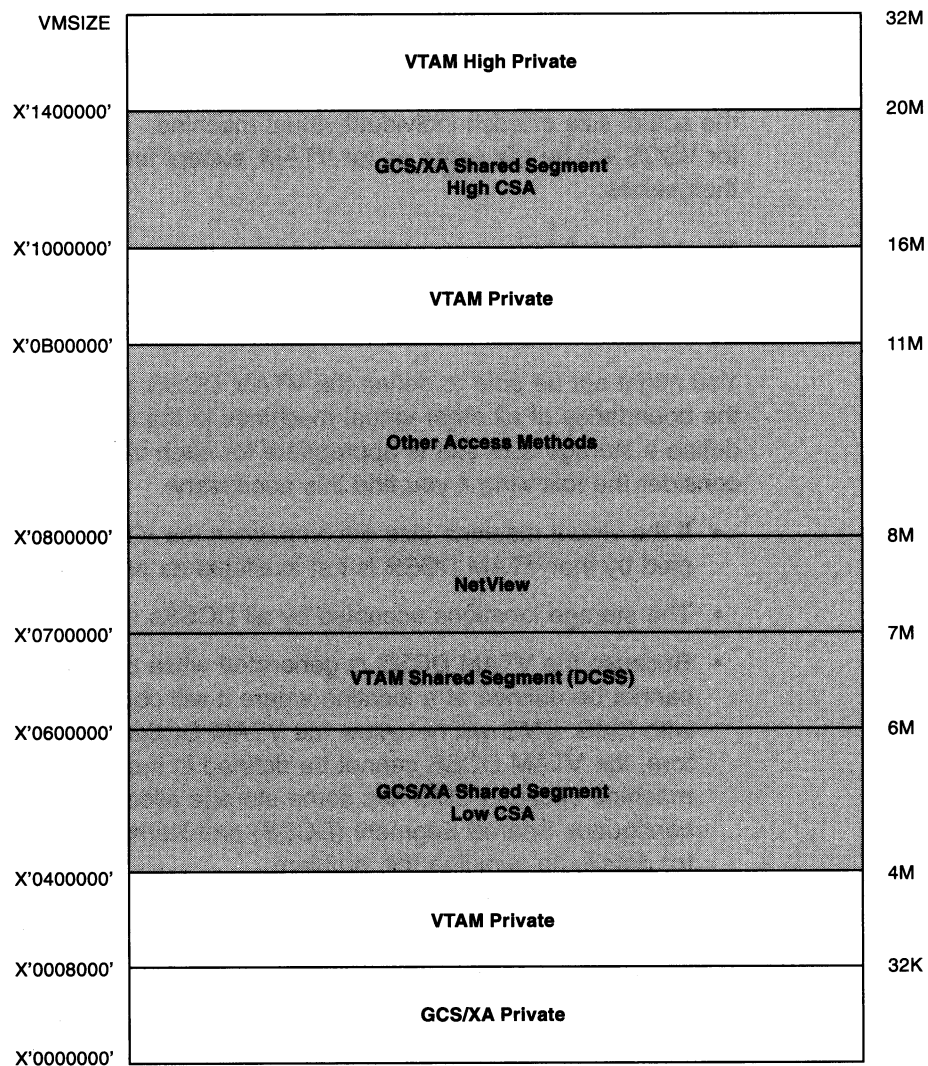


Figure 17. Sample Storage Layout for VM/ESA VTAM Virtual Machine

To define your VTAM DCSS, you need to find enough contiguous space for it. At least 1M is required, and the segment range X'600'–X'6FF' is recommended.

You can use the segment mapping tool, VMFSGMAP, to verify that this space is free for your VTAM DCSS. After logging on to the installation ID (5654010A) and establishing write access to the VMSES/E and software inventory disks, use the following VMFSGMAP command to use the segment mapping tool:

```
vmfsgmap segbld esasegs segblist
```

For additional information, refer to “Place VTAM into Production” on page 76.

Refer to the *VM/ESA GCS Reference* for details about GCS storage layout.

All GCS segments must be defined to start and end on 1M boundaries. For example, you cannot define a segment as X'F00'–X'F0F'; you must set it as X'F00'–X'FFF'.

Preparing Your Operating System

You must set up at least three virtual machines to install and run VTAM:

- 5654010A virtual machine is used to install and maintain VTAM. This virtual machine is referred to as the installation ID.
- VTAM virtual machine is where VTAM runs. This virtual machine is used for day-to-day operation of VTAM. It can have any user ID, but the name VTAM is recommended.
- GCS recovery virtual machine must be the first virtual machine started in the group.

You can also set up other virtual machines, such as:

VSCS if you want to run VSCS in its own virtual machine.

AUTOLOG1

if you want to have the GCS recovery machine started automatically as the first machine in the GCS group (as required).

To set up the virtual machines, you must do the following:

1. Use the DISKMAP command to determine where the minidisks are currently allocated and to search for free minidisk storage space. For example, if your directory is USER DIRECT, you should enter the command:

```
DISKMAP USER DIRECT
```

This creates the file USER DISKMAP. This file contains a list of the current addresses of the minidisks and the free storage space available. Use this information to map your space allocations in USER DIRECT. To find the sizes of the minidisks, see "5654010A Directory Entry."

2. Add the 5654010A and VTAM entries to USER DIRECT to define the virtual machines. The VM directory entries are located in the file USER DIRECT.

For detailed information about VM directory entries, refer to *VM/ESA Planning and Administration*.

5654010A Directory Entry

The 5654010A virtual machine directory entry must specify ownership of several minidisks required for VTAM installation and service. Table 4 lists these disks (including the 193 disk, which is owned by MAINT), and the address and size of each. You must modify the 5654010A directory entry to add the minidisks required by VTAM. These disks must be defined to VM before you install VTAM. Since this is the first release that uses VMSES/E, Table 4 contains the old names of the disks used in prior releases, as well as the new names.

Table 4 (Page 1 of 2). DASD Storage Requirements for Target Minidisks

Default Address	3380 Cyls	Description
2B2	75	Contains all the base code shipped with VTAM V4R2 for VM/ESA. Old Name=BASE New Name=BASE1

Table 4 (Page 2 of 2). DASD Storage Requirements for Target Minidisks

Default Address	3380 Cyls	Description
2C2	5	Samples disk. Old Name=VTM191 New Name=LOCALSAM
2C4	2	Contains customization files. This disk can also be used for local customer modifications. ¹ Old Name=ZAP New Name=LOCALMOD
2D2	20	Product service disk. ¹ Old Name=DELTA New Name=DELTA
2A6	10	Contains AUX files and software inventory tables that represent the test service level of VTAM V4R2 for VM/ESA. After testing, copy to 2A2. ¹ Old Name=MERGE New Name=APPLY TEST/ALTERNATE
2A2	10	Contains AUX files and software inventory tables that represent the service level of VTAM V4R2 for VM/ESA that is currently in production. ¹ Old Name=MERGE New Name=APPLY PRODUCTION
49A	40	Test build disk for VTAM server. After testing, copy to 29A. Contains load libraries and execs. Old Name=RUN New Name=BUILD0 TEST/ALTERNATE
29A	40	Production build disk for VTAM server. Old Name=RUN New Name=BUILD0 PRODUCTION
402	3	Test build disk for online OS/2 tools files. After testing, copy to 401. New Name=BUILD4 TEST/ALTERNATE
401	3	Production build disk for online OS/2 tools files. New Name=BUILD4 PRODUCTION
493	4	Test build disk for DVF, TRACERED, and formatted trace code. After testing, copy to MAINT's 193 disk. ³ Old Name=TRACERED/DVF New Name=BUILD2 TEST/ALTERNATE
193	4	Production copy of 493 disk. Old Name=TRACERED/DVF New Name=BUILD2 PRODUCTION
191	20	5654010A user ID's 191 minidisk.
Note: - The space requirements for these disks are initial recommended sizes. You may need to increase the disk space according to your service needs. - Cylinder values defined in this table are based on a 1k block size. - The space allocation for the MAINT 193 disk is necessary only for installing VTAM. This allocation should not be used as the maximum minidisk size because other products might also require space on the same minidisk.		

Following is a sample 5654010A directory entry:

```
USER 5654010A password 16M 64M EG
ACCOUNT 1 SYSPROG
IPL CMS
:
MDISK 191 3380 0969 020 VMSRES WR RMAINT WMAINT MMAINT
MDISK 2C2 3380 1009 005 VMSRES WR RMAINT WMAINT MMAINT
MDISK 2A6 3380 1014 010 VMSRES WR RMAINT WMAINT MMAINT
MDISK 2A2 3380 1024 010 VMSRES WR RMAINT WMAINT MMAINT
MDISK 2D2 3380 1095 020 VMSRES WR RMAINT WMAINT MMAINT
MDISK 2C4 3380 1115 002 VMSRES WR RMAINT WMAINT MMAINT
MDISK 401 3380 1117 003 VMSRES WR RMAINT WMAINT MMAINT
MDISK 402 3380 1120 003 VMSRES WR RMAINT WMAINT MMAINT
MDISK 493 3380 1123 004 VMSRES WR RMAINT WMAINT MMAINT
MDISK 193 3380 1127 004 VMSRES WR RMAINT WMAINT MMAINT
MDISK 2B2 3380 1131 075 VMSRES WR RMAINT WMAINT MMAINT
MDISK 29A 3380 1206 040 VMSRES WR RMAINT WMAINT MMAINT
MDISK 49A 3380 1246 040 VMSRES WR RMAINT WMAINT MMAINT
```

Note: The sample 5654010A directory entry shows space allocated on the volume ID VMSRES. These are sample entries; the space applies to a 3380 volume. You must modify the volume ID, device type, cylinder counts and ACCESS passwords for your environment.

VTAM Directory Entry

The VTAM virtual machine directory entry must specify the size of the VTAM virtual machine. You will need to update or create the directory entry for the VTAM virtual machine.

Following is a sample VTAM directory entry:

```
USER VTAM password 32M 32M ACEG
OPTION DIAG98 MAXCONN 400
MACHINE XA
IUCV *CCS P M 10
IUCV ANY P M 0
NAMESAVE gcssystem
ACCOUNT VTAM GCS
IPL gcssystem PARM AUTOLOG
CONSOLE 01F 3215 T OPERATOR
LINK 5654010A 2C2 191 RR
LINK 5654010A 29A 29A RR
```

Notes:

1. Notice that 2C2 is linked as 191. This enables 5654010A's 2C2 minidisk (LOCALSAM) to be automatically accessed as VTAM's A-disk.
2. Note that 29A is used in the production environment. If running in a test environment, you will need to make the appropriate directory updates for the VTAM and GCS user entries. The LINK statement for 29A should appear as follows:
LINK 5654010A 49A 29A RR
3. VTAM must be an XA-mode virtual machine, and the recommended size is 32M.
4. MAXCONN defines the number of IUCV connections that can be active concurrently. VSCS requires each terminal that is using VM to have an IUCV con-

nection. VSCS does not initiate the IUCV connection immediately when a user logs onto VSCS unless the logon request contains user data:

- If the request does not contain user data, VSCS displays the local VM system's logo screen only. The IUCV connection is not made until the user clears the logo screen by some action.
- If the request contains user data, VSCS immediately initiates an IUCV connection for the user.

VSCS uses one IUCV connection and GCS uses one IUCV connection. Therefore, you should specify MAXCONN as the number of users that will access VM through VSCS running in this virtual machine plus two (for VSCS and GCS).

5. It is recommended that you use the DIAG98 option. See Appendix C, "Tuning VM for VTAM" on page 649 for more information on DIAG98.
6. You can define the VTAM user ID as a privilege class ACEG user (as shown in the example). Class A permits you to issue CP commands such as SET QUICKDSP, SET SHARE RELATIVE, and ENABLE SNA from the VTAM virtual machine. Class C or E will enable you to use TRSOURCE and TRSAVE.
7. The IUCV ANY control statement enables VTAM to communicate with CP. The IUCV ANY control statement is required for all virtual machines in the GCS group, including the GCS recovery machine.
8. The IUCV *CCS control statement enables VSCS to communicate with CP. This statement is required for VSCS operation.

To add VTAM's ID and its associated options to VM, use DIRECTXA. Make sure you have accessed the disk on which the DIRECTXA directory resides, then issue the DIRECTXA command to place the updated directory online:

```
DIRECTXA USER DIRECT
```

You also use the DIRECT or DIRECTXA commands if you need to make any changes to VTAM's ID.

After you have created the directory entries and have run the DIRECT or DIRECTXA commands, run the DISKMAP command to check your DASD allocations. Resolve any overlaps.

Enabling Directory Changes

After you make all directory changes for the 5654010A user ID, you must log off the 5654010A user ID, then log back on to enable the directory changes. After logging back on, all the changes will be in effect.

Checkpointing APPN Data Files

The following data files are used when VTAM is defined as a network node or interchange node and are required for the APPN directory checkpointing and topology database checkpointing functions:

- DSDBCTRL DATA *
- DSDB1 DATA *
- DSDB2 DATA *
- TRSDB DATA *

These data files cannot be allowed to span multiple volumes. **DSDB1** and **DSDB2** hold APPN directory information. This information is used to initialize the directory

database when VTAM is restarted. Directory database information is stored alternately between DSDB1 and DSDB2. The directory database information is written to one of the data files whenever a MODIFY CHKPT TYPE=ALL or TYPE=DIR, HALT, or HALT QUICK command is issued.

Not all of the resources from the directory database are written to the data files when there is a checkpoint. The resources that are written to the data files are those that:

- Have been the target of a search
- Have a dynamic entry type that is not registered
- Have been updated within a period of time specified by the DIRTIME start option.

The resources that are registered to the database at start up through resource registration and definition are not included in the checkpointed information.

DSDBCTRL contains the current status of DSDB1 and DSDB2. It is read by VTAM during initialization to determine whether DSDB1 or DSDB2 will be used to load the APPN directory database.

TRSDDB is required for checkpointing the network topology database. The information in this data file is used to initialize the network topology database whenever VTAM is restarted. The network topology database is written to this file whenever a MODIFY CHKPT TYPE=TOPO or TYPE=ALL, HALT, or HALT QUICK command is issued.

These data files should be created prior to VTAM initialization. For your convenience, the FILEDEF statements needed to define these files have been included in the sample VMVTAM startup exec. They are commented out and you can uncomment them if you wish to use them.

If you do not use the sample statements provided, do the following to prepare the APPN checkpointing data files:

- Specify the FILEDEF name for DSDB1 DATA * as DSDB1, for DSDB2 DATA * as DSDB2, for DSDBCTRL DATA * as DSDBCTRL, and TRSDDB DATA * as TRSDDB. You can change the VM file name but not the FILEDEF name.
- Specify the following DCB subparameters for DSDB1 DATA, DSDB2 DATA, and TRSDDB DATA:
RECFL=FB,LRECL=1000,BLOCK=1000,DSORG=PS
- Specify the following DCB subparameters for DSDBCTRL DATA:
RECFL=FB,LRECL=20,BLOCK=20,DSORG=PS

Note: We recommend that you not modify any of these data files. We also recommend that you not copy the directory data files and attempt to use them during initialization of a different node. VTAM fails the initial load of the network topology database if another node's checkpointed data file is used or if the SSCPNAME operand is changed between the two IPLs. If the initial load fails, VTAM can acquire the information dynamically through topology database updates (TDUs).

Profile Requirements for Automatic Logon (AUTOLOG)

The VM AUTOLOG facility is used to log on the GCS group recovery machine. Using VM AUTOLOG ensures that the recovery machine is started first, a GCS requirement. If you want your virtual machines to be automatically started at system IPL time, then you will need to code profiles for your virtual machines.

The following examples illustrate the commands required to start VTAM. You can also add other commands. Please check Appendix C, "Tuning VM for VTAM" on page 649 for specific commands.

For details on the commands in this section, see the following:

- Appendix C, "Tuning VM for VTAM" on page 649
- *VTAM Operation* for the VTAM START and VSCS START commands
- *VM/ESA GCS Reference* for the GLOBAL and LOADCMD commands
- *VM/ESA Operator's Guide* for any other commands
- *VM/ESA CP Planning and Administration Guide* for any other commands.

This section details the profile requirements for the following virtual machines:

- AUTOLOG1
- GCS Recovery
- VTAM.

In a CMS environment, the PROFILE must have a filetype of EXEC. In a GCS environment, the PROFILE must have a filetype of GCS.

PROFILE for the AUTOLOG1 Virtual Machine

The AUTOLOG1 virtual machine automatically starts the GCS recovery machine.

Following is a sample PROFILE EXEC:

```
/* */
CP AUTOLOG recovery password
CP LOGOFF
```

AUTOLOG

Starts the GCS recovery virtual machine, where *recovery* is the name of your GCS recovery virtual machine.

LOGOFF Terminates the AUTOLOG1 virtual machine. The AUTOLOG1 virtual machine is not needed after the GCS recovery machine is started.

PROFILE for the GCS Recovery Virtual Machine

The GCS recovery virtual machine starts the VTAM virtual machine. The GCS recovery machine must be the first machine started in the GCS group.

Following is a sample PROFILE GCS:

```
/* */
CP AUTOLOG VTAM password
CP SET SHARE RELATIVE 10000
CP SET QUICKDSP
```

AUTOLOG

Starts the VTAM virtual machine. *password* must match the password coded on the USER control statement in the VTAM directory entry.

SET SHARE RELATIVE 10000

Improves the performance of GCS during error recovery.

SET QUICKDSP

Improves the performance of GCS during error recovery.

Note that the sample VTAM directory entry under "VTAM Directory Entry" on page 57 automatically causes an IPL GCSXA when VTAM is started (IPL GCSXA PARM AUTOLOG).

PROFILE for the VTAM Virtual Machine

Following is a sample PROFILE GCS for the virtual machine in which VTAM runs. Note that the installation process will overwrite this profile. If you wish to modify the profile, then change it after the installation process.

```
CP SET SHARE RELATIVE 10000
CP SET QUICKDSP
CP SET TIMER REAL
FILEDEF NCPLOAD DISK NCP LOADLIB A
VMVTAM
```

SET SHARE RELATIVE 10000

SET SHARE RELATIVE 10000 enables you to control the percentage of system resources that a virtual machine receives. These system resources include the processor (CPU cycles), real storage, and paging I/O capability. The RELATIVE operand specifies the relative share of the system assigned to a virtual machine. The maximum value that can be specified is 10000. Therefore, the VTAM virtual machine is allocated the maximum portion of total system resources available. It is recommended that you specify SET SHARE RELATIVE 10000.

SET QUICKDSP

The use of the QUICKDSP operand causes the VTAM virtual machine to be added to the dispatching list immediately when it has work to perform, without waiting in the eligible list. It is recommended that you use SET QUICKDSP.

SET TIMER REAL

The use of SET TIMER REAL causes the timer to be updated during the virtual processor run time and also during virtual wait time.

FILEDEF Defines the NCP load library (NCP LOADLIB A) on the VTAM 191 disk. This example assumes that the DDNAME=NCPLOAD on the NCP's BUILD statement.

The loader channel I/O trace file (LDRIOTAB) receives communication controller channel information if an NCP load fails. The information collected includes channel controller words, channel status words, and the first 20 bytes of any data associated with a WRITE, WRITEIPL, or WRITEBRK channel command.

If you want to use this trace, you must set up a file to receive the dumps. Add the following FILEDEF command to the sample PROFILE GCS above, and update the GLOBAL LOADLIB (in the sample VMVTAM startup EXEC) to identify the load module required for loading the NCP (NCP SSPGCS).

```
FILEDEF LDRIOTAB DISK filename type mode
```

If you are loading NCP and the LDRIOTAB file is not defined and you have an I/O error, the I/O trace table will then be dumped into storage under the heading I/O TRACE TABLE.

Refer to the *NCP, SSP, and EP Diagnosis Guide* for more information about the loader channel I/O trace file.

VMVTAM By default, the sample VMVTAM startup EXEC contains the following:

```
ACCESS 29A F/F
GLOBAL LOADLIB VTAM VSCS
LOADCMD VTAM ISTINV00
LOADCMD VSCS DTISLCMD
num='C555-555-5555'
pw='4394-0080-1454-3481-8008'
:
VTAM START CUSTNO='num',VTAMPW='pw',LIST=01,...
:
VSCS START
:
```

ACCESS Accesses the BUILD0 minidisk. This disk contains load libraries for VTAM and VSCS.

GLOBAL LOADLIB

Identifies to the VTAM virtual machine the load module libraries required for VTAM and VSCS. In this example, VSCS is running in the virtual machine with VTAM.

LOADCMD VTAM

Loads the VTAM command module and establishes *VTAM* as the prefix to be used for operator commands.

LOADCMD VSCS

Loads the VSCS command module and establishes *VSCS* as the prefix to be used for operator commands.

VTAM START

Starts VTAM in the virtual machine. A customer number (CUSTNO) and VTAM password (VTAMPW) must be included in the order shown as the first parameters on the VTAM start command. The CUSTNO and VTAMPW provided in the sample VMVTAM startup EXEC can be used to enable the VTAM Client/Server package. However, IBM recommends use of your own CUSTNO and VTAMPW; substitute your CUSTNO and VTAMPW for C555-555-5555 and 4394-0080-1454-3481-8008 respectively. To use VTAM MultiDomain or VTAM InterEnterprise, you must use your own CUSTNO and VTAMPW. For more information on the different VTAM packages, see the *VTAM Overview for VM/ESA and VSE/ESA*.

LIST=01 specifies the configuration list to be used (ATCCON01).

You may also include other parameters after these on the START command, or you can use a start list. See "Using Start Options and Configuration Lists" on page 127 for more information.

This command completes when VTAM initialization is complete and VTAM is ready to accept new commands.

VSCS START

Starts VSCS in the virtual machine.

Note: The CP ENABLE SNA command is issued automatically by VSCS. A DTIGEN option is available to turn off ENABLE SNA, if desired. However, without the ENABLE SNA command, VTAM will hang during initialization and not respond to VTAM commands. The VTAM operator must have class A, B, or F privileges to issue this command.

You might also want to consider the following when preparing the VTAM machine to execute its PROFILE:

- The VM AUTOLOG facility is used to log on the GCS group recovery machine. Using VM AUTOLOG ensures that the recovery machine is started first, a GCS requirement.
- VM's single console image facility (SCIF) is used to let the VM operator control VTAM from the system console.
- ATTACH commands give the VTAM machine control of channel-attached devices
- The VSCS PRINTER command initializes the table of printers available for the VM COPY PF key function.

You can also include AUTOLOG commands in the VTAM PROFILE to log on other machines in the GCS group.

Discontiguous Shared Segment (DCSS) and Named Saved Systems

A discontiguous shared segment is a saved segment that occupies one or more architected segments. It can be embedded above the virtual machine's defined storage size. A DCSS begins and ends on a megabyte boundary and is accessed by name. Though individual address ranges are specified on page boundaries anywhere within an architected segment, the DCSS is contained within megabyte boundaries. For more information on discontiguous shared segments, see *VM/ESA: Planning and Administration*.

GCS consists of a named, shared segment in storage that you can IPL and run in a virtual machine. It is a virtual machine supervisor that bands many virtual machines together in a group and supervises their operations. GCS also provides an interface between program products such as VTAM. For more information on GCS, see *VM/ESA: General Information*.

For information on defining a DCSS for VTAM and a named saved system for GCS, see "Defining Discontiguous Shared Segment (DCSS) and Named Saved Systems" on page 82.

Loading VTAM - Overview

The install process uses the VMFINS INSTALL command. The *VTAM Program Directory* that is shipped with VTAM describes the procedure for installing VTAM from the distribution tape including all control files. When the VMFINS INSTALL command is issued, the following steps occur:

- The first tape file is loaded from the product tape.
- The System Inventory requisite and description tables are updated.
- The requisites are checked.

If the requisites are not satisfied, a list of missing products are displayed and the option to continue is provided. If you want to continue, VMFINS loads the product to the system even though requisites are missing.

- The Software Inventory is checked to see if the product is already installed.
- VTAM's current installation parameters may be overridden on the MAKE OVERRIDE PANEL.
- System resources are then allocated and generated.
- The product is installed.
- The Software Inventory is updated to reflect that the files have been received and the product is installed.

For complete installation instructions, see "Installation Instructions" on page 66. For more information on the VMFINS command, refer to the *VM/ESA VMSES/E Introduction and Reference*.

Figure 18 on page 65 illustrates the process of loading and maintaining VTAM under VM. For more information on the referenced disks, see Table 4 on page 55.

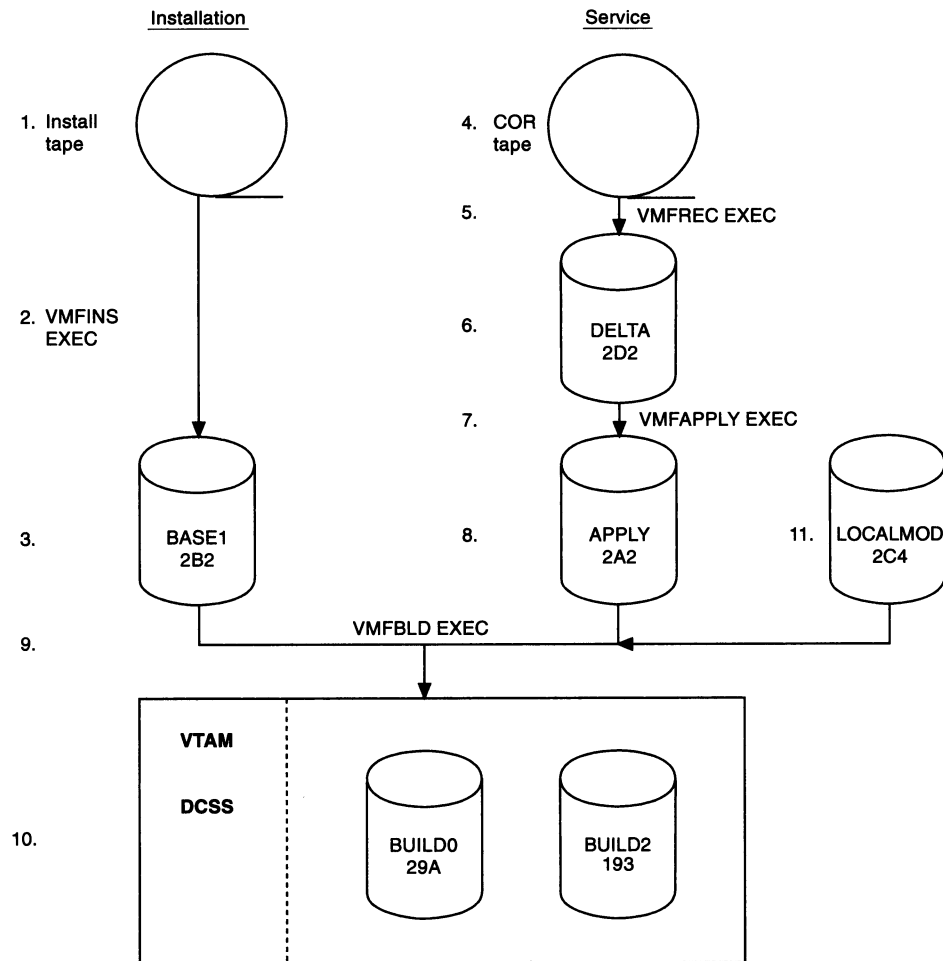


Figure 18. Installing and Maintaining VTAM under VM

Following are the steps illustrated in Figure 18:

1. Mount and attach the installation tape to the 5654010A installation ID.
2. Run the VMSES/E installation procedure, VMFINS.
3. The contents of the tape are unloaded onto the BASE1 2B2 disk.
4. You might have service (COR tape or program temporary fixes) to apply to your system.
5. Use the VMFREC EXEC to receive service from the service tape onto the DELTA 2D2 disk.
6. The DELTA disk is the PTF staging disk.
7. Use the VMFAPPLY EXEC to update your APPLY disk with the new service.
8. The APPLY disk is used to handle merged service. It contains all the files that are new or have been changed since VTAM was installed.
9. Build VTAM using the VMFBLD EXEC.
10. VTAM runs from the system (BUILD0) disk.
11. If you have any local maintenance that you want to apply to VTAM, you must update the LOCALMOD 2C4 disk before building VTAM. For more information

on local modifications using VMSES/E, refer to the *VMSES/E Introduction and Reference*.

Installation Instructions

This section describes the step-by-step procedures to install and activate VTAM V4R2 for VM/ESA. This section duplicates the installation instructions in the *VTAM Program Directory*. Note, however, that the *VTAM Program Directory* might contain more accurate information that was not available at the time this book was produced.

The step-by-step procedures are in two column format. The steps to be performed are in bold large numbers. Commands for these steps are on the left side of the page in bold print. Additional information for a command exist to the right of the command. For more information about the two column format, refer to “Understanding Dialogs with the System” in *VM/ESA Installation (SC24-5526)*.

Each step of the installation instructions must be followed. Do not skip any step unless directed otherwise.

Throughout these instructions, the use of IBM-supplied default minidisk addresses and user IDs is assumed. If you use different user IDs or minidisk addresses to install VTAM V4R2 for VM/ESA, adapt these instructions as needed for your environment.

Refer to *VMSES/E Introduction and Reference* for details on the use and syntax of the VMSES/E commands you use to install VTAM V4R2 for VM/ESA.

Note: The sample console output presented throughout these instructions was produced on a VM/ESA Version 1 Release 2.2 system. If you are installing VTAM V4R2 for VM/ESA on a different VM/ESA system, the results obtained for some commands might differ from those depicted here.

Installation of VTAM V4R2 for VM/ESA with VMSES/E (VMFINS)

You use VMFINS to install VTAM V4R2 for VM/ESA. VMFINS is an installation exec supplied as part of VMSES/E to make installation of VM and Licensed Program Products (LPs) consistent.

For a complete description of all VMFINS installation options refer to *VMSES/E Introduction and Reference (SC24-5444)*.

VMSES/E Installation Process Overview

The following is a brief description of the main steps in installing VTAM V4R2 for VM/ESA using VMSES/E:

- Plan your installation for VTAM V4R2 for VM/ESA.
Use the VMFINS command to load several VMSES/E files from the product tape to obtain VTAM V4R2 for VM/ESA resource requirements. For more information, see “Plan Your Installation” on page 67.
- Allocate resources for installing VTAM V4R2 for VM/ESA.
Use the information obtained from the previous step to allocate the appropriate minidisks and user IDs needed to install and use VTAM V4R2 for VM/ESA. For

more information, see “Allocate Resources for Installing VTAM V4R2 for VM/ESA” on page 71.

- Install the VTAM V4R2 for VM/ESA product.

Use the VMFINS command to load the VTAM V4R2 for VM/ESA product files from tape to the appropriate test minidisks and directories. For more information, see “Install VTAM V4R2 for VM/ESA” on page 71.

- Update the build status table.

Use VMFINS to update the VM SYSBLDS file used by VMSES/E for software inventory management.

For more information, see “Update Build Status Table for VTAM V4R2 for VM/ESA” on page 74.

- Copy VTAM V4R2 for VM/ESA files into production.

Once the product files have been tailored and the operation of VTAM V4R2 for VM/ESA is satisfactory, the product files are copied from the test BUILD disks to production BUILD. For more information, see “Copy VTAM Files Into Production” on page 81.

- Activate VTAM V4R2 for VM/ESA.

For information about initial activation of VTAM V4R2 for VM/ESA, see “Activate the Program” on page 81.

- Install VTAM VIT analysis tool

If you will be using the VTAM VIT analysis tool, you will need to perform some additional steps for installation of this function. For more information, see “Installing the VTAM VIT Analysis Tool” on page 96.

- Install OS/2 code provided by VTAM.

Once VTAM V4R2 for VM/ESA has been installed, download and unpack the VTAM-provided OS/2 DLUR and the VTAM Command Set Library. For more information, see Appendix L, “Communications Manager/2's DLUR Support” on page 705 and “Installing the VTAM Command Set Library for IBM Command Tree/2” on page 104, respectively.

For a complete description of all VMSES/E installation options refer to *VMSES/E Introduction and Reference* (SC24-5444).

Plan Your Installation

You use the VMFINS command to plan the installation. This is a two step process that:

- Loads the first tape file, containing installation files
- Generates a “PLANINFO” file listing containing:
 - All user ID and minidisk requirements
 - All the required products.

To obtain planning information for your environment:

- 1** Log on as the installer or planner.

This user ID can be **any** ID that has read access to MAINT's 5E5 minidisk and write access to MAINT 51D minidisk.

- 2** Mount the VTAM V4R2 for VM/ESA installation tape and attach it to the user ID at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.

- 3** Establish read access to the VMSES/E code.

link maint 5e5 5e5 rr
access 5e5 b

VMSES/E resides on the 5E5 disk.

- 4** Establish write access to the Software Inventory disk.

link maint 51d 51d mr
access 51d d

The VMSES/E system-level Software Inventory and other dependent files reside on the MAINT 51D disk.

Note: If another user is currently linked to the MAINT 51D minidisk in write mode (R/W), you will obtain only read access (R/O). If this occurs, you need to have that user re-link to the 51D in read-only mode (RR), and then re-issue the above LINK and ACCESS commands. Do not continue with these procedures until you establish a R/W link to the 51D minidisk.

5 Load the VTAM V4R2 for VM/ESA product control files to the 51D minidisk.**vmfins install info (nomemo**

The NOMEMO option loads the memo from the tape but does not issue a prompt to send the memo to the system printer. Specify the MEMO option if you want to be prompted to print the memo.

The INSTALL INFO command performs the following:

- Loads the Memo-to-Users (contains a pointer to the *VTAM Program Directory* and this *VTAM Network Implementation Guide*)
- Loads various product control files, including the Product Parameter File (PPF) and the PRODPART files
- Creates VMFINS PRODLIST on your A-disk. The VMFINS PRODLIST contains a list of products on the installation tape.

When complete, VMFINS returns the following:

Sample VM/ESA V1R2.2 console output

```
vmfins install info (nomemo
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS1909I VMFINS PRODLIST created on your A-disk
VMFINS2760I VMFINS processing completed successfully
Ready;
```

End of Sample VM/ESA V1R2.2 console output

6 Obtain resource planning information for VTAM V4R2 for VM/ESA.

vmfins install ppf 5654010A VTAM (plan nomemo)

The PLAN option specifies that VMFINS performs requisite checking, plans system resources, and provides an opportunity to override the defaults in the product parameter file (PPF). VTAM will **not** be loaded at this time using this form of the VMFINS command.

You can override any of the following:

- The name of the product parameter file (PPF)
- The default user IDs
- The minidisk/directory definitions

Notes:

- If you change the PPF name, a default user ID, or any other parameters via a PPF override, you need to use your changed values instead of those indicated (when appropriate), throughout the rest of these installation instructions, as well as the instructions provided for servicing VTAM V4R2 for VM/ESA. For example, you will need to specify your PPF override file name instead of 5654010A for certain VMSES/E commands.
- If you are not familiar with creating PPF overrides using VMFINS, see "Using the Make Override Panel" section in Chapter 3 of the *VMSES/E Introduction and Reference* before you continue.

Sample VM/ESA V1R2.2 console output

```
vmfins install ppf 5654010a vtam (plan nomemo
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2601R Do you want to create an override for :PPF 5654010A VTAM :PRODID 5654010A%VTAM?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINS2603I Processing product :PPF 5654010A VTAM :PRODID 5654010A%VTAM
VMFREQ2805I Product :PPF 5654010A VTAM :PRODID 5654010A%VTAM has passed requisite checking
VMFINT2603I Planning for the installation of product :PPF 5654010A VTAM :PRODID 5654010A%VTAM
VMFRMT2760I VMFRMT processing started
VMFRMT2760I VMFRMT processing completed successfully
VMFINS2760I VMFINS processing completed successfully
Ready;
```

End of Sample VM/ESA V1R2.2 console output

- 7** Review the install message log (\$VMFINS \$MSGLOG). All install message logs are written to the installation user ID's A-disk. If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview install

Allocate Resources for Installing VTAM V4R2 for VM/ESA

Use the planning information in the 5654010A PLANINFO file, created in the **PLAN** step (section "Plan Your Installation" on page 67) to create the 5654010A user directory for minidisk install.

- 1** Obtain the user directory from the 5654010A PLANINFO file.

Note: The user directory entry is located at the bottom of the PLANINFO file of the resource section; these entries will contain all of the links and privilege classes necessary for the 5654010A user ID. Use the directory entry found in PLANINFO as a model as input to your system directory.

- 2** Add the MDISK statements to the directory entry for 5654010A. Use Table 4 on page 55 to obtain the minidisk requirements.
- 3** Add the 5654010A directory to the system directory. Change the passwords for 5654010A from xxxxx to a valid password, in accordance with your security guidelines.
- 4** Place the new directories online using VM/Directory Maintenance (DIRMAINT) or an equivalent CP directory maintenance method.

Note: All minidisks for the 5654010A user ID must be formatted before installing VTAM V4R2 for VM/ESA.

Install VTAM V4R2 for VM/ESA

- 1** Logon to the installation user ID **5654010A**.
- 2** Create a PROFILE EXEC that contains the ACCESS commands for MAINT 5E5 and 51D minidisks.

```
xedit profile exec a
==> input /**/
==> input 'access 5e5 b'
==> input 'access 51d d'
==> file
```

- 3** Execute the profile to access MAINT's minidisks.

profile

- 4** Establish write access to the Software Inventory disk, if it is not already linked R/W.

Note: If the MAINT 51D minidisk was accessed R/O, you need to have the user who has it linked R/W re-access the disk as R/O. You then can issue the following commands to establish R/W access.

```
link maint 51d 51d mr
access 51d d
```

- 5** Have the VTAM V4R2 for VM/ESA installation tape mounted and attached to 5654010A at virtual address 181. The VMFINS EXEC requires the tape drive to be at virtual address 181.
- 6** Install VTAM V4R2 for VM/ESA.

Notes:

If you have already created a PPF override file, you should specify your override file name after the **ppf** keyword for the following VMFINS command.

You might be prompted for additional information during VMFINS INSTALL processing depending on your installation environment. If you are unsure how to respond to a prompt, refer to the “Installing Products with VMFINS” and “Install Scenarios” chapters in the *VMSES/E Introduction and Reference* to decide how to proceed.

vmfins install ppf 5654010A VTAM (nomemo nolink

The NOLINK option indicates that VMFINS is not to link to the appropriate minidisks, only access them if not accessed.

Sample VM/ESA V1R2.2 console output (1 of 2)

```

vmfins install ppf 5654010a vtam (nomemo nolinek
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2601R Do you want to create an override for :PPF 5654010A VTAM :PRODID 5654010A%VTAM?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINS2603I Processing product :PPF 5654010A VTAM :PRODID 5654010A%VTAM
VMFREQ2805I Product :PPF 5654010A VTAM :PRODID 5654010A%VTAM has passed requisite checking
VMFINT2603I Installing product :PPF 5654010A VTAM :PRODID 5654010A%VTAM
VMFSET2760I VMFSETUP processing started for 5654010A VTAM
VMFUTL2205I Minidisk|Directory Assignments:
      String      Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E    R/W  2C2   DMR2C2
VMFUTL2205I LOCALMOD  F    R/W  2C4   DMR2C4
VMFUTL2205I APPLY     G    R/W  2A6   DMR2A6
VMFUTL2205I           H    R/W  2A2   DMR2A2
VMFUTL2205I DELTA     I    R/W  2D2   DMR2D2
VMFUTL2205I BUILD0    J    R/W  49A   DMR49A
VMFUTL2205I BUILD2    K    R/W  493   DMR493
VMFUTL2205I BUILD4    L    R/W  402   DMR402
VMFUTL2205I BASE1     M    R/W  2B2   DMR2B2
VMFUTL2205I -----  A    R/W  191   USR191
VMFUTL2205I -----  B    R/O  5E5   MNT5E5
VMFUTL2205I -----  C    R/O  292   ISP192
VMFUTL2205I -----  D    R/W  51D   MNT51D
VMFUTL2205I -----  S    R/O  190   MNT190
VMFUTL2205I -----  Y/S   R/O  19E   MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFREC2760I VMFREC processing started
VMFREC1852I Volume 1 of 1 of INS TAPE 9500
VMFREC1851I (1 of 9) VMFRCAXL processing AXLIST
VMFRCX2159I Loading 0 part(s) to DELTA 2D2 (I)
VMFREC1851I (2 of 9) VMFRCPTF processing PARTLST
VMFRCP2159I Loading 0 part(s) to DELTA 2D2 (I)
VMFREC1851I (3 of 9) VMFRCCOM processing DELTA
VMFRC2159I Loading 0 part(s) to DELTA 2D2 (I)
VMFREC1851I (4 of 9) VMFRCALL processing APPLY
VMFRC2159I Loading part(s) to APPLY 2A6 (G)
VMFRC2159I Loaded 1 part(s) to APPLY 2A6 (G)
VMFREC1851I (5 of 9) VMFRCALL processing BASE
VMFRC2159I Loading part(s) to BASE1 2B2 (M)
VMFRC2159I Loaded 3339 part(s) to BASE1 2B2 (M)
VMFREC1851I (6 of 9) VMFRCALL processing SAMPLE
VMFRC2159I Loading part(s) to LOCALSAM 2C2 (E)
VMFRC2159I Loaded 18 part(s) to LOCALSAM 2C2 (E)

```

End of Sample VM/ESA V1R2.2 console output (1 of 2)

Sample VM/ESA V1R2.2 console output (2 of 2)

```
VMFREC1851I (7 of 9) VMFRCALL processing BUILD
VMFRCA2159I Loading part(s) to BUILD0 49A (J)
VMFRCA2159I Loaded 5 part(s) to BUILD0 49A (J)
VMFREC1851I (8 of 9) VMFRCALL processing TRACE
VMFRCA2159I Loading part(s) to BUILD2 493 (K)
VMFRCA2159I Loaded 26 part(s) to BUILD2 493 (K)
VMFREC1851I (9 of 9) VMFRCALL processing WSCODE
VMFRCA2159I Loading part(s) to BUILD4 402 (L)
VMFRCA2159I Loaded 4 part(s) to BUILD4 402 (L)
VMFREC2760I VMFREC processing completed successfully
VMFINT2603I Product installed
VMFINS2760I VMFINS processing completed successfully
Ready;
```

End of Sample VM/ESA V1R2.2 console output (2 of 2)

- 7** Review the install message log (\$VMFINS \$MSGLOG). All install message logs are written to the installation user ID's A-disk. If necessary, correct any problems before going on. For information about handling specific error messages, refer to *VM/ESA: System Messages and Codes*, or use online HELP.

vmfview install

Update Build Status Table for VTAM V4R2 for VM/ESA

- 1** Update the VM SYSBLDS software inventory file for VTAM V4R2 for VM/ESA.

vmfins build ppf 5654010A VTAM (serviced nolink

The SERVICED option builds any parts that were not built on the installation tape (if any) and updates the Software Inventory build status table showing that the product 5654010A has been built.

Sample VM/ESA V1R2.2 console output

```

vmfins build ppf 5654010a vtam (serviced nolink
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2603I Processing product :PPF 5654010A VTAM :PRODID 5654010A%VTAM
VMFREQ2805I Product :PPF 5654010A VTAM :PRODID 5654010A%VTAM has passed
requisite checking
VMFINB2603I Building product :PPF 5654010A VTAM :PRODID 5654010A%VTAM
VMFSET2760I VMFSETUP processing started for 5654010A VTAM
VMFUTL2205I Minidisk|Directory Assignments:
      String      Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E    R/W   2C2   DMR2C2
VMFUTL2205I LOCALMOD  F    R/W   2C4   DMR2C4
VMFUTL2205I APPLY     G    R/W   2A6   DMR2A6
VMFUTL2205I          H    R/W   2A2   DMR2A2
VMFUTL2205I DELTA     I    R/W   2D2   DMR2D2
VMFUTL2205I BUILD0     J    R/W   49A   DMR49A
VMFUTL2205I BUILD2     K    R/W   493   DMR493
VMFUTL2205I BUILD4     L    R/W   402   DMR402
VMFUTL2205I BASE1     M    R/W   2B2   DMR2B2
VMFUTL2205I -----  A    R/W   191   USR191
VMFUTL2205I -----  B    R/O   5E5   MNT5E5
VMFUTL2205I -----  C    R/O   292   ISP192
VMFUTL2205I -----  D    R/W   51D   MNT51D
VMFUTL2205I -----  S    R/O   190   MNT190
VMFUTL2205I -----  Y/S   R/O   19E   MNT19E
VMFSET2760I VMFSETUP processing completed successfully
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I No new build requirements identified
VMFBLD2179I There are no build requirements matching your request at this time.
           No objects will be built
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
VMFINB2603I Product built
VMFINB2173I No verification exec found for this product
VMFINS2760I VMFINS processing completed successfully
Ready;

```

End of Sample VM/ESA V1R2.2 console output

Place VTAM into Production

Define and Build the VTAM V4R2 for VM/ESA Saved Segments Using VMSES/E

You should build segments for 5654010A. First, you define the segments to the system using the segment mapping tool VMFSGMAP. Once the segments are defined, use VMFBLD to build them. For more information, see “Defining Discontiguous Shared Segment (DCSS) and Named Saved Systems” on page 82.

For more information on using VMSES/E for saved segments, review the chapter, “Using VMSES/E to Define, Build, and Manage Saved Segments” in *VM/ESA Planning and Administration*.

Note: The defining and building of the VTAM V4R2 for VM/ESA saved segments should be performed from the installation user ID. If you move any segments that are currently defined on your system you, must ensure that they are rebuilt from the user ID that maintains them.

- 1** Logon to the installation user ID **5654010A**.
- 2** Establish write access to the Software Inventory disk.

**link maint 51d 51d mr
access 51d d**

- 3** Add VTAM V4R2 for VM/ESA segment object definitions to the SEGBLIST EXC00000 build list.

vmfsgmap segbld esasegs segblist

This command displays a panel for making segment updates. See Figure 19 on page 77 for an example of the Segment Map panel.


```

VMFSGMAP - Segment Map                                     More: +
Lines 1 to nn of nn

      000-MB          001-MB          002-MB          003-MB
Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
M CMS   SYS W-W-----1.....2.....3.....
M GCS   SYS W-----1.....2.....3.....

      004-MB          005-MB          006-MB          007-MB
Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
CMSPIPES DCS 4.....5.....6.....RRRR-----
M GCS   SYS RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNN6.....7.....
M HLASM DCS 4.....5.....RRRRRRRRRRRRRRR7.....

      008-MB          009-MB          00A-MB          00B-MB
Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
DOSBAM SPA 8.....9.....A.....=====
CMSBAM MEM 8.....9.....A.....BRRR.....
CMSDOS MEM 8.....9.....A.....R.....
CMSVMLIB DCS RRRRRRRRRRRRRRRR9.....A.....B.....
DOSINST DCS 8.....R-----A.....B.....

      00C-MB          00D-MB          00E-MB          00F-MB
Name   Typ 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
HELPINST DCS RRRRRRRRRRRRRRRRD.....E.....F.....
M CMS   SYS C.....D.....RRRRRRRRRRRRRRRRRRRRRRRRRRRRRR>

===== 16-MB Line =====

      010-MB          011-MB          012-MB          013-MB
F1=Help    F2=Chk Obj  F3=Exit    F4=Chg Obj  F5=File    F6=Save
F7=Bkwd    F8=Fwd       F9=Retrieve F10=Add Obj F11=Del Obj F12=Cancel

=====> _

```

Figure 19. Segment Map panel example.

4 Select the Add Object panel by pressing PF10.

F10

F10 takes you from the Segment Map panel to the Add Segment Definition panel. See Figure 20 on page 78 to see an example of the Add Segment Definition panel.

Add Segment Definition

Lines 1 to *nn* of *nn*

```

OBJNAME..... VTAM
DEFPARMS....:
SPACE.....:
TYPE.....: SEG
OBJDESC.....:
OBJINFO.....:
GT_16MB..... NO
DISKS.....:
SEGREQ.....:
PRODID.....: 5654010A
BLDPARMS....: UNKNOWN
  
```

F1=Help
F2=Get Obj
F3=Exit
F4=Add Line
F5=Map
F6=Chk MEM

F7=Bkwd
F8=Fwd
F9=Retrieve
F10=Seginfo
F11=Adj MEM
F12=Cancel

====>

Figure 20. Add Segment Definition Panel Example

5 Obtain the VTAM V4R2 for VM/ESA segment definitions from the PRODPART file by pressing F10.

OBJNAME..... VTAM
PRODID.....: 5654010A

F10

F10 obtains the VTAM V4R2 for VM/ESA segment information from the 5654010A PRODPART file. See Figure 21 for the refreshed Add Segment definition panel that will be displayed.

Add Segment Definition

More: +
Lines 1 to *nn* of *nn*

```

OBJNAME..... VTAM
DEFPARMS....: 600-6FF SR
SPACE.....:
TYPE.....: SEG
OBJDESC.....: VTAM TEST SEGMENT
OBJINFO.....: MUST BE BELOW 16M LINE....
GT_16MB..... NO
DISKS.....:
SEGREQ.....:
PRODID.....: 5654010A VTAM
BLDPARMS....: PPF(5654010A VTAM ISTSBSEG)
  
```

VMFSMD2760I Seginfo processing completed SUCCESSFULLY

F1=Help
F2=Get Obj
F3=Exit
F4=Add Line
F5=Map
F6=Chk MEM

F7=Bkwd
F8=Fwd
F9=Retrieve
F10=Seginfo
F11=Adj MEM
F12=Cancel

====>

Figure 21. Add Segment Definition Panel Showing the Segments

6 Return to the Segment Map panel by pressing F5.

F5

F5 returns you to the Segment Map panel. See Figure 22 for an example of the refreshed Segment Map panel.

VMFSGMAP - Segment Map										More: -	
										Lines nn to nn of nn	
Meg		004-MB			005-MB			006-MB			007-MB
St Name	Typ	0123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF
M GROUP1	SPA	4.....	5.....	6.....	7.....	8.....	9.....				
M NETVSG00	MEM	4.....	5.....	6.....	7.....	8.....	9.....				
M NPM	MEM	4.....	5.....	6.....	7.....	8.....	9.....				
P VTAM	DCS	4.....	5.....	6.....	7.....	8.....	9.....				
M ISRDCCS	DCS	4.....	5.....	6.....	7.....	8.....	9.....				
M GCS	SYS	RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	6.....	7.....	8.....	9.....					
M GCSXA	SYS	RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	6.....	7.....	8.....	9.....					
M GCSXA1	SYS	RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	6.....	7.....	8.....	9.....					
M GCSXA2	SYS	RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	6.....	7.....	8.....	9.....					
M GCSXA3	SYS	RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	6.....	7.....	8.....	9.....					
M GCSXA4	SYS	RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	6.....	7.....	8.....	9.....					
Meg		008-MB			009-MB			00A-MB			00B-MB
St Name	Typ	0123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF
M DOSBAM	SPA	8.....	9.....								
M CMSDOS	MEM	8.....	9.....								
M CMSBAM	MEM	8.....	9.....								
M CMSAMS	MEM	8.....	9.....								
M CMSVSAM	MEM	8.....	9.....								
M ISPDCCS	DCS	8.....	9.....								
M CMSVLIB	DCS	RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	9.....								
M CMSPIPES	DCS	8.....	9.....								
M DOSINST	DCS	8.....	9.....								
Meg		00C-MB			00D-MB			00E-MB			00F-MB
St Name	Typ	0123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF0	123456789	ABCDEF
M HELPINST	DCS	RRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN									
M CMS	SYS	C.....	D.....								
===== 16-MB Line =====											

Figure 22. Segment Map panel

7 Save the new information and exit from the Segment Map panel by pressing F5.

F5

F5 saves all changed information and exits the map panel.

8 Prepare to build the segments. This is a three step process.

a IPL CMS to clear the virtual storage

ipl cms parm clear nosprof instseg no

**** DO NOT press ENTER at the VMREAD!****

This command allows you to bypasses the execution of the system profile (SYSPROF EXEC) without loading the installation saved segment (CMSINST).

access (nosprof

Bypass the execution of the PROFILE EXEC.

b Access the VMSES/E code

access 5e5 b

c Establish write access to the Software Inventory disk

**link maint 51d 51d mr
access 51d d**

9 Issue VMFBLD command to build the VTAM V4R2 for VM/ESA segments.

vmfblld ppf segbld esasegs segblist VTAM (ALL

Sample VM/ESA V1R2.2 console output

```
VMFBLD2760I VMFBLD processing started
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I New build requirements identified
VMFBLD1851I (1 of 1) VMFBDSEG processing SEGBLIST EXC00000, target is BUILD 51D (D)
VMFBDS2115I Validating segment VTAM
VMFBDS2002I A DEFSEG command will be issued for 1 segment(s).
VMFBDS2219I Processing object VTAM.SEGMENT
HCPNSS440I Saved segment VTAM was successfully saved in fileid 0028.
VMFBLD1851I (1 of 1) VMFBDSEG completed with return code 0
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
Ready;
```

End of Sample VM/ESA V1R2.2 console output

Activate the Program

Perform the following steps to ensure that VTAM V4R2 for VM/ESA was installed properly:

- 1** See "Preparing Your Operating System" on page 55 to prepare your operating system prior to activating VTAM.
- 2** Logon to your VTAM user ID **VTAM**.
- 3** Detach the BUILD0 PRODUCTION (29A) disk and link to the installation user ID's BUILD0 TEST/ALTERNATE (49A) and LOCALSAM (2C2) disks.

```
detach 29a
link 5654010a 49a 29a rr
link 5654010a 2c2 2c2 rr
```

- 4** IPL GCSXA

```
i gcsxa
```

- 5** Access the LOCALSAM (2C2) disk.

```
access 2c2 a
```

- 6** Bring up VTAM using the sample VMVTAM startup EXEC.

```
vmvtam
```

Message IST0201I tells you that VTAM is operational.

- 7** To end VTAM, enter VTAM HALT.

```
vtam halt
```

Copy VTAM Files Into Production

- 1** Logon to the installation user ID **5654010A**.
- 2** Access the BUILD0 TEST/ALTERNATE (49A) disk and the BUILD0 PRODUCTION (29A) disk.

```
access 49a e
access 29a f
```

- 3** Issue the VMFCOPY command to copy all the files from the BUILD0 TEST/ALTERNATE (49A) disk to the BUILD0 PRODUCTION (29A) disk. Note that message VMFCOP2866I will be received indicating that file VMSES PARTCAT will not be copied.

```
vmfcopy * * e = f2 (prodid 5654010A%VTAM olddate replace
```

- 4 Access the BUILD4 TEST/ALTERNATE (402) disk and the BUILD4 PRODUCTION (401) disk.

access 402 e
access 401 g

- 5 Issue the VMFCOPY command to copy all the files from the BUILD4 TEST/ALTERNATE (402) disk to the BUILD4 PRODUCTION (401) disk.

vmfcopy * * e = = g2 (prodid 5654010A%VTAM olddate replace

- 6 Access the BUILD2 TEST/ALTERNATE (493) disk and the BUILD2 PRODUCTION (193) disk.

access 493 e
access 193 h

- 7 Issue the VMFCOPY command to copy all the files from the BUILD2 TEST/ALTERNATE (493) disk to the BUILD2 PRODUCTION (193) disk.

vmfcopy * * e = = h2 (prodid 5654010A%VTAM olddate replace

VTAM V4R2 for VM/ESA is now installed and built on your system.
--

Install VTAM VIT Analysis Tool

If you will be using the VTAM VIT analysis tool, you will need to perform some additional steps for installation of this function. For more information, see "Installing the VTAM VIT Analysis Tool" on page 96.

Install OS/2 Code

Once you have installed VTAM V4R2 for VM/ESA, download and unpack the VTAM-provided OS/2 DLUR and the VTAM Command Set Library. For more information, see Appendix L, "Communications Manager/2's DLUR Support" on page 705 and "Installing the VTAM Command Set Library for IBM Command Tree/2" on page 104.

Defining Discontiguous Shared Segment (DCSS) and Named Saved Systems

You must define a DCSS for VTAM and a named saved system for GCS using VMSES/E.

Note: Although the DEFSEG and SAVESEG commands will still allow you to define and save the VTAM DCSS, it is strongly recommended that you use the

VMSES/E execs and commands. This will keep the VMSES/E software inventory files up to date.

Prior to building the DCSS for VTAM, refer to *Planning and Defining CP Saved Segments* and *Using VMSES/E to Define, Build and Manage Saved Segments* in *VM/ESA Planning and Administration* for further information. Then follow the steps below. See "Define and Build the VTAM V4R2 for VM/ESA Saved Segments Using VMSES/E" on page 76 for additional information on these steps, such as examples of the panels.

1. VMSES/E provides a mapping tool called the VMFSGMAP EXEC. This exec allows you to plan the allocation of space for a DCSS and named saved systems, in relation to other defined segments.

To invoke the VMFSGMAP tool, issue the following command:

```
VMFSGMAP SEGBLD ESASEGS SEGBLIST
```

VMFSGMAP contains many options, one of which is *Add Obj*. The VTAM DCSS can be added to the virtual machine using this option. When *Add Obj* is used, the *Add Segment Definition* panel is displayed. On this panel, first enter VTAM in the *OBJNAME* field and 5654010A in the *PRODID* field. Then press F10 and VTAM provides defaults for the remaining fields. After that, press F5 to map the segment, and a refreshed main panel is displayed that includes the newly defined VTAM DCSS. Pressing F5 again files your changes.

For additional information on the VMFSGMAP EXEC, see the *VMSES/E Introduction and Reference*.

Notes:

- a. VTAM now occupies a full meg of storage and will no longer have the space available to share a one-meg segment with another segment. Prior to this release, it was possible to pack other segments with VTAM within the same meg of storage. VTAM and another segment can both be members of a larger segment space.
 - b. If the default DEFPARMS (X'600'-X'6FF') for the VTAM DCSS are not observed, care must be taken not to define a member saved segment with VTAM that can possibly corrupt the layout of virtual machine storage.
2. The VTAM segment must be loaded under the 16M line. You should ensure that the VTAM DCSS does not overlap other segments within the GCS group. Although VTAM does not run with CMS, the VTAM DCSS is generated under the control of CMS. Consequently, when applying maintenance and regenerating the VTAM DCSS, IPL the CMS session with the following commands:

```
IPL CMS PARM CLEAR NOSPROF INSTSEG NO (Do not hit enter at the VM READ prompt!)
ACC (NOPROF
LINK MAINT 5E5 5E5 RR (link and access the VMSES/E code on 5E5 and 51D)
LINK MAINT 51D 51D MR (note the access in write mode)
ACC 5E5 B
ACC 51D D
```

3. The following VMFBLD command allows for the VTAM DCSS segment to be built.

```
VMFBLD PPF SEGBLD ESASEGS SEGBLIST VTAM (ALL
```

Note: VMFBLD processing includes the equivalent of processing that, in previous releases, took place after issuing the DEFSEG VTAM 600-6FF SR

command. VMFBLD also loads the DCSS with VTAM's code, which used to occur when the VTAM installation process issued a CP SAVESEG command.

During a successful VMFBLD, the following message is displayed:

```
HCPNSS440I Saved segment VTAM was successfully saved in fileid 0028.
```

In the above message text, the fileid is a spool number that you can query with the CP QUERY NSS command. The segment that you just defined is a class S spool file, which is a skeleton DCSS that can be loaded. The VMSES/E installation process defines, loads, and saves the VTAM DCSS.

If HCPNSS440I is not displayed the new segment was not built, unless there was already a VTAM saved segment prior to this build that did not have any new service applied to it. The (*SERVICED* option of the VMFBLD EXEC only rebuilds if service has been added.

For an explanation of any VTAM error messages received, see *VTAM Messages and Codes*.

If you define the GCS named saved system or the VTAM DCSS as restricted, then you must include a NAMESAVE entry in the directory of each virtual machine that will access the named saved system or the DCSS. Because GCSXA loads all the DCSS's defined using the GROUP exec at GCS IPL time, the NAMESAVE entries must be in each virtual machine that IPLs GCS. For more information on the NAMESAVE option, see *VM/ESA Planning and Administration*.

The VMFBLD should be issued during the installation and during the service. When you apply service, if there is a part that was on the list as being serviced and it is also a part of the segment, you will need to issue the VMFBLD command for the segment. For complete service instructions, see "Applying Service" on page 86.

If you need to redefine the DCSS, remember that the old segment will need to be purged with the VMFSGMAP (either *Delete Object* or *Change Object*). See the *VM/ESA Planning and Administration*.

Generating Group Control System (GCS)

The group control system (GCS) provides supervisor and data management services for the VTAM virtual machine and other virtual machines in which VTAM application programs run. VTAM does not run unless GCS has been generated and is operational.

Before you generate GCS, you must configure it. To configure GCS, use the GROUP exec and do the following:

- Define an authorized virtual machine to receive dumps. This user ID must also be specified in the list of authorized users. If you do not do this, you will receive partial dumps when an error occurs.
- Define the VTAM virtual machine as an authorized machine.
- If VSCS operates in a virtual machine other than the VTAM virtual machine, define the VSCS virtual machine as an authorized machine.
- Define the GCS recovery virtual machine for the GCS group. This virtual machine must be different from the VTAM virtual machine.

- Specify the maximum number of virtual machines that can exist in the GCS group.
- Specify the size of the GCS trace table.

For information about the GROUP exec, refer to *VM/ESA Planning and Administration*, or the *VM/ESA GCS Reference*.

Verifying the Installation

The verification procedure is based on the following assumptions:

- The VTAM user ID default disk addresses have not been modified. (If they have been modified, you might have to modify the sample VTAM start-up EXEC.)
- The VTAM default directory entry is used.
- GCS has been installed and verified.
- The GCS recovery machine is logged on.
- The VTAM machine is an authorized GCS machine.

To verify the installation, complete the following:

1. Log on to the VTAM user ID. Enter:

```
LOGON VTAM password
```

2. IPL GCS:

```
IPL gcssystem
```

3. If you defined your own set of VSCS start options or coded any VTAM user-exit routines, issue the following command to enable your modifications:

```
GLOBAL LOADLIB VSCSUSER VTAMUSER VSCS VTAM
```

4. Access the 2C2 disk.

5. To start VTAM use VMVTAM, the sample VTAM startup EXEC. Enter VMVTAM from the VTAM ID.

6. You will receive a series of messages regarding VTAM and VSCS initialization. Following is a sample of the messages you will receive:

```
GCTACC423I F (029A) R/O
IST1497I VTAM FUNCTIONAL SUPPORT LEVEL IS INTERENTERPRISE
IST093I ISTCDRDY ACTIVE
IST495I SIZE HAS BEEN SET TO 050
IST315I VTAM INTERNAL TRACE ACTIVE - MODE = INT, SIZE = 050
IST199I OPTIONS = NONE
IST314I END
IST093I ISTDSWMN ACTIVE
IST093I ISTAPPLS ACTIVE
IST020I VTAM INITIALIZATION COMPLETE FOR V4R2
IST1349I COMPONENT ID IS 5654-01001-420
IST1348I VTAM STARTED AS SUBAREA NODE
DTII10I DEFAULT INITIALIZATION PARAMETERS BEING USED
DTIC01I DTICINIT VSCS IUCV INITIALIZATION IS IN PROGRESS
DTIC17I DTICINIT ENABLE SNA ISSUED
Command complete
DTIS74I TRACE TABLE STARTS AT 00800020 END 0080FA5F , CURRENT 00800040 , SIZE 2000
DTIC02I DTICINIT VSCS IUCV INITIALIZATION COMPLETE
Ready;
DTIV04I DTIVINIT VSCS VTAM SERVICES INITIALIZATION COMPLETED
IST984I USER EXIT ISTEXCSD IS ACTIVE
```

Notes:

- a. Message IST020I tells you that VTAM is installed and operational.
 - b. Message DTIV04I tells you that VSCS is installed and operational.
 - c. The SIZE value for message IST315I might be different.
 - d. The OPTIONS value for message IST199I might be different.
 - e. The ADDRESSES for message DTIS74I might change.
 - f. If you receive message IST004I, verify that the BUILD0 disk is accessed and the segment is built correctly.
 - g. If you receive message IST116I or IST054I, verify that the 2C2 LOCALSAM disk is accessed.
7. To terminate VSCS and VTAM, enter:
- VTAM HALT

Applying Service

Installing Tables and Modules in VM

When making local modifications to your system, use VMSES/E commands and execs in place of VMFLKED as in prior releases. Complete instructions are available in the *VM/ESA: VMSES/E Introduction and Reference* and the *VM/ESA: Service Guide*.

VTAM User Modifications during Installation:

To make local modifications to your system during installation, follow the steps below:

1. Create or copy the IBM-supplied assemble file needed for your local modification. See Table 5 on page 87 for the appropriate part and member name. This is required in rebuilding any local modifications you have put on your system.
2. Copy the assemble file to the 2C4 (E-disk). Make your local modification to the copy on your LOCALMOD 2C4 disk.
3. Issue the following command:

```
GLOBAL MACLIB VTAMBLD VTAMAC
```
4. Issue the assemble command for the file:

```
VMFHASM fn 5654010A VTAM (PPF
```

Notes:

- a. Other options are available for the assemble commands. Consult the *VM/ESA: VMSES/E Introduction and Reference* for additional information.
 - b. We recommend that you use the VMFHASM assemble exec supplied by VMSES/E. Also available is the VMFHLASM assemble exec.
 - c. If the assemble function is successful, the file *fn* TXT00000 will be placed on the A-disk.
5. Copy the *fn* TXT00000 file from the A-disk to the 2C4 E-disk as *fn* TXTLnnnn.

Note: TXTL is a required filetype for local modified text. The *nnnn* is a user-defined number assigned to this fix, usually starting with 0001.

6. Erase the TXT00000 from your A-disk.
7. Rename the assemble file on the 2C4 E-disk as *fn* ASML0001 E.
8. Update the VVT tables for both the TXT and ASM files by issuing the following VMFSIM against both the text and the assemble files:


```
VMFSIM LOGMOD 5654010A VVTCL E TDATA :MOD LCL0001 :PART fn TXT
VMFSIM LOGMOD 5654010A VVTCL E TDATA :MOD LCL0001 :PART fn ASM
```
9. Relink your new local modification into the appropriate loadlib by issuing the following command. See Table 5 for the appropriate *blist* and *memname*.


```
VMFBLD PPF 5654010A VTAM blist memname (ALL
```
10. Issue the following command after GCS has been IPLed but before VTAM is started:


```
GLOBAL LOADLIB VSCSUSER VTAMUSER VSCS VTAM
```

Table 5 (Page 1 of 2). Local Modifiable Assemble Files Provided by IBM

FILENAME (fn)	Buildlist (blist)	Member name (memname)	LOADLIB affected
ISTINCNO	ISTBLVL1	ISTINCNO	VTAM
ISTINCLM	ISTBLVL1	ISTINCLM	VTAM
ISTINCL1	ISTBLVL1	ISTTABLE	VTAM
ISTINCL2	ISTBLVL1	DSIASCII	VTAM
ISTINCL3	ISTBLVL1	DSIASCPR	VTAM
ISTINCL4	ISTBLVL1	DSICNMDT	VTAM
ISTINCL5	ISTBLVL1	DSILU0	VTAM
ISTINCL6	ISTBLVL1	DSIXDOM	VTAM
ISTINCL7	ISTBLVL1	DSI4LU2	VTAM
ISTINCL8	ISTBLVL1	DSI6LU2	VTAM
ISTINCL9	ISTBLVL1	TABBINDS	VTAM
ISTMGC10	ISTBLVL1	ISTMGC10	VTAM
ISTINCDT	ISTBLVL1	ISTINCDT	VTAM
DTISDMOD	ISTBLVL2	DTIISTR	VSCS
DTIPRLOG	ISTBLVL2	DTIPRLOG	VSCS
DTIPDADV	ISTBLVL2	DTIPDADV	VSCS
DTISMSGM	ISTBLVL2	DTIISTR	VSCS
DTIUSERn	ISTBLVL8	DTIUSERn	VSCSUSER
ISTEXCAA	ISTBLVL9	ISTEXCAA	VTAMUSER
ISTEXCVR	ISTBLVL9	ISTEXCVR	VTAMUSER
ISTPUCWC	ISTBLVL9	ISTPUCWC	VTAMUSER
ISTEXCCS	ISTBLVL9	ISTEXCCS	VTAMUSER
ISTEXCSD	ISTBLVL9	ISTEXCSD	VTAMUSER
ISTCMMND	ISTBLVL9	ISTCMMND	VTAMUSER

Table 5 (Page 2 of 2). Local Modifiable Assemble Files Provided by IBM

FILENAME (fn)	Buildlist (blist)	Member name (memname)	LOADLIB affected
ISTEXCUV	ISTBLVL9	ISTEXCUV	VTAMUSER
ISTEXCDM	ISTBLVL9	ISTEXCDM	VTAMUSER
ISTMSFLD	ISTBLVL9	ISTMSFLD	VTAMUSER

Service Instructions

This section contains the procedure to install CORrective service to VTAM V4R2 for VM/ESA. This section duplicates the service instructions in the *VTAM Program Directory*. Note, however, that the *VTAM Program Directory* may contain more accurate information that was not available at the time this book was produced.

VMSES/E is used to install service for VTAM V4R2 for VM/ESA. To become more familiar with service using VMSES/E, read the introductory chapters in *VMSES/E Introduction and Reference* (SC24-5444). This manual also contains the command syntax for the VMSES/E commands listed in the procedure. Also see the *VM/ESA Service Guide* (SC24-5527).

Note: Each step of the servicing instructions must be followed. Do not skip any step unless otherwise directed. All instructions showing accessing of disks assume the use of default minidisk addresses. If different minidisk addresses are used, change the instructions appropriately.

VMSES/E Service Process Overview

The following is a brief description of the main steps in servicing VTAM V4R2 for VM/ESA using VMSES/E:

- Merging service

Use the VMFMRDSK command to clear the alternate apply disk before receiving new service. This allows you to easily remove the new service if a serious problem is found.
- Receiving service

The VMFREC command receives service from the delivery media and places it on the Delta disk.
- Applying service

The VMFAPPLY command updates the version vector table (VVT), which identifies the service level of all the serviced parts. In addition, AUX files are generated from the VVT for parts that require them.
- Reapplying local service (if applicable)

All local service (mods) must be entered into the software inventory to allow VMSES/E to track the changes and build them into the system. See Chapter 7 in the *VM/ESA Service Guide* (SC24-5527) for this procedure.
- Building new levels

The build task generates the serviced level of an object and places the new object on a test BUILD disk.
- Placing the new service into production

Once the service is satisfactorily tested it should be put into production by copying the new service to the production disk, re-saving the NSS (Named Saved System) or DCSS (Discontiguous Saved Segments), and so on.

Servicing VTAM V4R2 for VM/ESA

Prepare to Receive Service

1 Log on to the VTAM V4R2 for VM/ESA service user ID **5654010A**.

2 Establish access to the software inventory disk.

Note: If the MAINT 51D minidisk was accessed R/O, you will need to have the user that has it accessed R/W link it R/O. You can then issue the following commands to obtain R/W access to it:

**link maint 51d 51d mr
access 51d d**

The 51D minidisk is where the VMSES/E Software Inventory files and other product dependent files reside.

3 Have the VTAM V4R2 for VM/ESA CORrective service tape mounted and attached to **5654010A**.

4 Establish the correct minidisk access order.

vmfsetup 5654010A VTAM

5654010A is the PPF that was shipped with the product. If you have your own PPF override, substitute your PPF name for 5654010A.

Electronic Service: If you are receiving service from Service Link (electronic service) see Appendix A, "Receiving COR Service from ServiceLink," in the *VM/ESA Service Guide* for the steps to receive service documentation. Return here to continue the service process.

5 Receive the documentation. VMFREC with the INFO option loads the documentation and displays a list of all the products on the tape.

vmfrec info

6 Check the receive message log (\$VMFREC \$MSGLOG) for warning and error messages.

vmfview receive

Also make note of which products and components have service on the tape. To do this, use the PF5 key to show all status messages which identify the products on the tape.

7 Clear the alternate APPLY disk to ensure that you have a clean disk for new service.

Applying Service

vmfmrdsd 5654010A VTAM apply

This command clears the alternate APPLY disk.

5654010A is the PPF that was shipped with the product. If you have your own PPF override you should substitute your PPF name for 5654010A.

- 8** Review the merge message log (\$VMFMRD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfvlew mrd

Receive the Service

- 1** Receive the service.

vmfrec ppf 5654010A VTAM

This command receives service from your service tape. All new service is loaded to the alternate DELTA disk.

vmfrec ppf 5654010A VTAM (env vptfnnnn

This command receives service from Service Link that now resides on disk.

This command maps the envelope file (verifies that all the necessary files exist), creates a SERVICE DISKMAP file, and receives service from your PTF envelope VPTFnxxx SERVLINK.

- 2** Review the receive message log (\$VMFREC \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfvlew receive

Apply the Service

- 1** Apply the new service.

vmfapply ppf 5654010A VTAM

This command applies the service that you just received. The version vector table (VVT) is updated with all serviced parts and all necessary AUX files are generated.

- 2** Review the apply message log (\$VMFAPP \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see *VM/ESA: System Messages and Codes*, or use online HELP.

vmfvlew apply

Update the Build Status Table**1** Update the Build Status Table with serviced parts.**vmfbld ppf 5654010A VTAM (status**

This command updates the Build Status Table.

Note

If the \$ppf files have been changed by the service you have applied, you will get the following prompt:

VMFBLD2185R The following source product parameter files have been serviced:

VMFBLD2185R 5654010A \$PPF

VMFBLD2185R When source product parameter files are serviced, all product parameter files built from them must be recompiled using VMFPPF before VMFBLD can be run.

VMFBLD2185R Enter zero (0) to have the latest levels of the source product parameter files copied to your A-disk and exit VMFBLD so you can recompile your product parameter files with VMFPPF
Enter one (1) to continue only if you have already recompiled your product parameter files with VMFPPF

0

Enter a 0 and complete the following steps before you continue.

VMFBLD2188I Building 5654010A \$PPF on 191 (A) from level \$PFnnnnn

VMFBLD2760I VMFBLD processing completed

Ready(00500);

vmfppf 5654010A VTAM

Note: If you have created your own PPF override, use your PPF name instead of 5654010A.

copyfile 5654010A \$ppf a = = d (olddate replace
erase 5654010A \$ppf a

Note: Do not use your own PPF name in place of 5654010A for the COPYFILE and ERASE commands.

vmfbld ppf 5654010A VTAM (status**1**

Re-issue VMFBLD to complete updating the build status table.

Note: If you have created your own PPF override, use your PPF name instead of 5654010A.

When you receive the prompt that was previously displayed, enter a 1 to continue.

Applying Service

- 2 Use VMFVIEW to review the build status messages, and see what objects need to be built.

vmfview build

Build Serviced Objects

- 1 Rebuild VTAM V4R2 for VM/ESA serviced parts.

vmfbld ppf 5654010A VTAM (serviced

- 2 Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see VM/ESA: System Messages and Codes, or use on-line HELP.

vmfview build

Place the New VTAM Service into Production

Rebuild the Saved Segments

- 1 Re-save the VTAM V4R2 for VM/ESA segments.

vmfbld ppf segbld esasegs segblist VTAM (serviced

- 2 Review the build message log (\$VMFBLD \$MSGLOG). If necessary, correct any problems before going on. For information about handling specific error messages, see VM/ESA: System Messages and Codes, or use on-line HELP.

vmfview build

Copy the New VTAM Serviced Files Into Production

- 1 Logon to the installation user ID **5654010A**.
- 2 Access the BUILD0 TEST/ALTERNATE (49A) disk and the BUILD0 PRODUCTION (29A) disk.

access 49a e
access 29a f

- 3** Issue the VMFCOPY command to copy all the files from the BUILD0 TEST/ALTERNATE (49A) disk to the BUILD0 PRODUCTION (29A) disk. Note that message VMFCOP2866I will be received indicating that file VMSES PARTCAT will not be copied.

vmfcopy * * e = = f2 (prodid 5654010A%VTAM olddate replace

- 4** Access the BUILD4 TEST/ALTERNATE (402) disk and the BUILD4 PRODUCTION (401) disk.

**access 402 e
access 401 g**

- 5** Issue the VMFCOPY command to copy all the files from the BUILD4 TEST/ALTERNATE (402) disk to the BUILD4 PRODUCTION (401) disk.

vmfcopy * * e = = g2 (prodid 5654010A%VTAM olddate replace

- 6** Access the BUILD2 TEST/ALTERNATE (493) disk and MAINT's 193 disk.

**link maint 193 193 mr
access 493 e
access 193 h**

- 7** Issue the VMFCOPY command to copy all the files from the BUILD2 TEST/ALTERNATE (493) disk to MAINT's 193 disk.

vmfcopy * * e = = h2 (prodid 5654010A%VTAM olddate replace

- 8** Access the APPLY TEST/ALTERNATE disk (2A6) disk and the APPLY PRODUCTION (2A2) disk.

**access 2A6 e
access 2A2 i**

- 9** Issue the VMFCOPY command to copy all files that reside on the APPLY TEST/ALTERNATE disk (2A6) disk to the APPLY PRODUCTION (2A2) disk.

vmfcopy * * e = = l2 (prodid 5654010A%VTAM olddate replace

You now have finished servicing VTAM V4R2 for VM/ESA.

Trouble Shooting Installation and Service Problems

Below is a list of possible problems you can run into during the installation process, along with possible reasons why they occurred and actions to correct them:

Problem: When building the VTAM DCSS the following error message is issued:

VMFBDS2012E Storage in range xxxxxxxx-xxxxxxx for VTAM in use.

Programmer Response: This message is issued when the installation execs are trying to load the VTAMDCSS module into virtual storage. The default address it is loading at is X'600000'–X'6FFFFFF'. This error message means that the virtual storage is already in use by CMS and the segment cannot be saved.

Use the Q SEGMENT command to determine if one of the CMS libraries is allocated where you want VTAM to be placed. If there are CMS libraries like the CMSVMLIB, you can issue SEGMENT RELEASE CMSVMLIB to drop the segments and rebuild the VTAM segment.

Problem: The following error message is issued when building VTAM:

DMSMOD109S Virtual Storage Capacity Exceeded:

Programmer Response: Check that your installation userid is defined with enough virtual storage to fully encompass the VTAM DCSS. For example, the segment is loaded from X'600000'–X'6FFFFFF' by default, in which case the installation ID cannot be defined any smaller than 7M.

We recommend that the installation ID has 16M of virtual storage. Issue the following CP command:

DEF STOR 16M

Problem: When verifying the VTAM installation, an ABEND 806 occurs when initializing VTAM:

GCTABD226E Application 'ISTINV01' failed - System abend 806-0004:

Programmer Response: Review the following for possible causes:

1. The VTAM DCSS was not successfully saved. Issue the following command:

Q NSS MAP NAME vtam

where *vtam* is the name of the VTAM DCSS defined using VMFSGMAP. The VTAM DCSS should have a class of A (Active). If the VTAM DCSS does not exist or has a class of S (Skeleton), review “Defining Discontiguous Shared Segment (DCSS) and Named Saved Systems” on page 82 to correctly build the VTAM DCSS.

2. The VTAM segment is not being loaded by GCS. Define the VTAM DCSS to GCS as an authorized segment (AUTHSEG).
3. The VTAM segment has been overlaid with another segment. Use VMFSGMAP to verify that the VTAM save segment does not overlap with another segment defined to GCS as an authorized segment (AUTHSEG).

If all of the above seem to be correct, perform the following detailed analysis:

1. Determine which segments are being loaded by GCS. GCS loads the segments indicated in the CONSEGS label in memory. Get the GCSNUC MAP file and search for the CONSEGS label in the listing. The entry will state the CONSEGS label is at location *x* and is *y* bytes large, where *x* is the storage address of CONSEGS and *y* is the length.

2. Log on to the VTAM machine and IPL GCS without starting up the VMVTAM exec.

Note: If your profile automatically invokes the PROFILE GCS exec, disable it.

3. Once GCS is IPL'ed, issue:

D TX.Y

The system will display output similar to the following, assuming *x* equals X'5770' and *y* equals X'20':

```
R00005770 E5E3C1D4 40404040 00600000 006FFFFFF *VTAM.....*
R00005780 D5C5E3E5 E2C7F0F0 00700000 007FFFFFF *NETVSG00.....*
```

This indicates VTAM is being loaded from X'600'–X'6FF' and NETVSG00 is being loaded from X'700'–X'7FF'.

If the CONSEGS entry does not show the VTAM segment name, an AUTHSEG statement needs to be supplied in the GCS GROUP file and GCS needs to be rebuilt.

If the NETVSG00 and VTAM entries both had the same address ranges as shown below, GCS would first load the VTAM segment and then overlay it with the NETVSG00 segment at the same location (in this case, X'600000'–X'6FFFFFF'):

```
R00005770 E5E3C1D4 40404040 00600000 006FFFFFF *VTAM.....*
R00005780 D5C5E3E5 E2C7F0F0 00600000 006FFFFFF *NETVSG00.....*
```

4. If the CONSEGS entry looks okay, the problem might be that the VTAM segment was not loaded. GCS loads the VTAM segment when an IPL GCS is done. Since this should have been done, display the VTAM segment and you should see the CONTENTS macro in storage as follows:

D T600000.60

```
R00600000 C3D6D5E3 C5D5E3E2 00000000 00000000 64 *CONTENTS.....*
R00600010 C3C3D2D9 C9E4E6E3 0060DAD8 00000000 *CCKRIUWT.-.Q....*
R00600020 C9E2E3C1 C9C3C1D9 0060B894 00000000 *ISTAICAR.-.m....*
R00600030 C9E2E3C1 C9C3C9C1 0060C36C 00000000 *ISTAICIA.-C%....*
R00600040 C9E2E3C1 C9C3C9D6 0060C8AC 00000000 *ISTAICIO.-H.....*
R00600050 C9E2E3C1 C9C3C9D9 0060CDC0 00000000 *ISTAICIR.-.....*
```

If you do not see the CONTENTS macro, the VTAM segment was not built successfully. Refer to "Defining Discontiguous Shared Segment (DCSS) and Named Saved Systems" on page 82 for information on how to rebuild the segment.

Problem: An ABEND 806 occurs in the NetView or FTP virtual machine during initialization.

Programmer Response: Use the VMFSGMAP command to verify that the CMSBAM and CMSVSAM segments do not overlap the VTAM DCSS. The NetView and FTP applications use VSAM files which require the use of the CMSBAM and CMSVSAM segments.

Problem: The following error message is issued when starting VTAM:

IST001I VTAM Start Rejected - GCS Authname Failed

Another symptom of this problem might be that VTAM dumps with S0C1.

Programmer Response: Ensure that the VTAM segment is defined as a full meg of storage and that the segment was saved correctly.

Problem: The VMFBLD command executes with a good return code but, when you try to IPL your GCSXA machine, you go into a program interrupt loop.

Programmer Response: Verify that you have not written over GCS with VTAM code by defining the VTAM segment where the GCSXA segment should be defined.

Communication Manager/2's DLUR Support

For information on Communication Manager/2's DLUR support, including information on downloading and unpacking the DLUR files, see Appendix L, "Communications Manager/2's DLUR Support" on page 705.

Installing the VTAM VIT Analysis Tool

To install the VTAM VIT analysis tool, complete the following steps as described in this section:

1. Access the appropriate disks
2. Update and run the ISPF EXEC
3. Install the ISPF trace tables
4. Invoke the ISPF Dialog Tag Language Utility
5. Compile the help panels, creating multiple help panels
6. Verify that the trace formatter panels have been set up correctly
7. Optionally customize the ISPF interface.

If problems are encountered, see "VTAM Internal Trace (VIT) Analysis Tool Problems (MVS) (VM)" in *VTAM Diagnosis* for further information.

Accessing the Appropriate Disks

The following loadlibs or maclibs contain the necessary data to set up the VTAM VIT Analysis Tool. You need to concatenate the maclibs into the ISPF EXEC, as shown below.

Target Loadlib or Maclib	Action	Mini Disk	Comment
N/A	Accessed as A	493 Test 193 Prod	Contains REXX EXECs
ISTPLIB	Accessed as A	493 Test 193 Prod	Contains compiled panels
ISTMLIB	Accessed as A	493 Test 193 Prod	Contains compiled ISPF messages
user-defined maclib	Accessed as A	493 Test 193 Prod	This maclib can be a new or existing one and needs to be the same maclib used for ISPTABL.
ISTTLIB (same as used for ISPTLIB)	Accessed as A	493 Test 193 Prod	Because ISPTABL can only point to one maclib, this maclib needs to replace any previous maclib set up for ISPTLIB.
ISTDEBUG	Accessed as B	49A Test 29A Prod	Contains ISTRAFT1 load module
N/A	Accessed as C	2B2 BASE1 disk	Contains input GML files

Updating and Running the ISPF EXEC

Update the ISPF EXEC with the appropriate maclib names and then run it. See Figure 23 on page 98 for an example. Note that the highlighted FILEDEF statements in Figure 23 on page 98 are required, and they must be placed after the ISRNULL FILEDEF statements.

Installing the ISPF Trace Tables

Figure 23. Sample ISPF EXEC Modified for IST Maclibs

After the installation maclibs have been concatenated and the mini disk for the table EXEC is accessed, you can install the ISPF tables. To do this, select option 7 from the ISPF/PDF PRIMARY OPTION MENU, and then option 1 from the Dialog Test menu to go to the ISPF INVOKE DIALOG FUNCTION/SELECTION PANEL shown in Figure 24 on page 99.

```

----- INVOKE DIALOG FUNCTION/SELECTION PANEL -----

INVOKE SELECTION PANEL:
  PANEL ===>                                OPT ===>

INVOKE COMMAND:
  CMD ===> ISTTTABL

  LANG ===>                                (APL or blank)

INVOKE PROGRAM/SHARED SEGMENT:
  PGM ===>                                DCSS ===>

  PARM ===>                                EXTENDED PLIST ===> NO

NEWAPPL      ===> NO                        ID      ===>

NEWPOOL      ===> NO                        PASSLIB  ===> NO

COMMAND ===>
F1=HELP      F2=SPLIT  F3=END    F4=RETURN  F5=RFIND  F6=RCHANGE
F7=UP        F8=DOWN   F9=SWAP   F10=LEFT  F11=RIGHT  F12=CURSOR

```

Figure 24. ISPF Panel Used to Run the Table EXECs

To run all the table EXECs for the formatted trace, enter the EXEC name ISTTTABL in the INVOKE COMMAND: CMD ===> field and press Enter.

The individual table EXECs for formatted trace are:

- ISTTT007
- ISTTT012
- ISTTT017
- ISTTT024.

Each table EXEC should complete with a return code of 0. RC=0 will appear in the upper right corner of the screen.

The output of the table EXECs is placed in the maclib pointed to by the ISPTABL statement, the user-defined ISPTLIB maclib.

Invoking the ISPF Dialog Tag Language Utility

To complete the panel setup, you must compile all of the help panels that expand into multiple panels. All other panels, help panels, keylists, and commands have been compiled and placed in the appropriate maclib. The ISPF Dialog Tag Language Utility, ISPD TLC, is used for the compile. ISPD TLC is a REXX EXEC provided by ISPF that generates panels, help panels, messages, keylists and commands from Dialog Tag Language source files. For more information about the Dialog Tag Language Utility, see the *ISPF Dialog Tag Language and Reference Guide*. You can enter ISPD TLC from the COMMAND ===> line or from the COMMAND AND EXEC PROCESSING panel, shown in Figure 25 on page 100.

```
ISRCMS----- COMMAND AND EXEC PROCESSING -----
COMMAND ==>

ENTER CMS OR CP COMMAND BELOW:
==> ISPD TLC

Any one of the following may be entered:

- A CMS command
- A CP command
- An EXEC specification
- SUBSET (to enter CMS subset mode: use RETURN command
          to terminate subset mode.)

F1=HELP   F2=SPLIT  F3=END    F4=RETURN  F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN   F9=SWAP   F10=LEFT  F11=RIGHT  F12=CURSOR
```

Figure 25. ISPF Panel Used to Invoke the Dialog Tag Language Utility

After ISPD TLC has been invoked, the ISPF Dialog Tag Language Conversion Utility panel is displayed.

Compiling the Help Panels, Keylists, and Commands

The following section describes how to compile the help panels, keylists, and commands using the ISPF Dialog Tag Language Conversion Utility.

To compile the help panels, keylist, and commands, enter the following on the "ISPF Dialog Tag Language Conversion Utility" panel:

- The GML source file. ISTTTWO contains a compile list of all the help panels that expand into multiple help panels.
- The output panel maclib for the panels.
- The output log for the messages.
- The four-character prefix of the trace application to be used as the keylist application ID. For formatted trace, enter ISTT.
- Whether old files are replaced with the new compiled output. Enter YES.
- Whether you want the ISPD TLC compiler messages displayed on the screen or written to a file.

If you indicate that you want the messages written to an output file, you can specify a filename in the OUTPUT LOG field. Otherwise, the output is written to the ISPF log.

Writing the messages to the ISPF log will not cause any messages to be overwritten. It is recommended that you have the messages written to a file.

- Whether you want ISPF messages suppressed. It is recommended that you not suppress messages.
- Your national language.

Examples of the completed ISPF V3R2 panels follow.

To compile the help panels, keylists, and commands for formatted trace, enter the information as provided on the screen shown in Figure 26.

```

                                ISPF Dialog Tag Language Conversion Utility

Enter requested information:
GML source file . . . . . ISTTTWO
Output Panel MACLIB fn . . 'ISTPLIB'
    (Leave blank for sequential panel file)
Output Log fn . . . . . _____
    (Leave blank to use ISPF log file)

Keylist Application Id . . ISTT Up to four characters

Replace old files . . . . . YES No or Yes
ISPD TLC messages to disk . YES No or Yes
Allow DBCS . . . . . NO No or Yes
Specify KANA . . . . . _____ No or Yes or blank
Suppress messages (ISPF) . NO No or Yes
National Language . . . . . ENGLISH

Command ==> _____

```

Figure 26. Compiling the Help Panels, Keylists, and Commands for Formatted Trace

After compiling the help panels and tables have been generated and/or moved to the correct maclibs, the VTAM VIT analysis tool is ready for use or testing. No additional steps are necessary.

Verifying the Trace Formatter Panels

To verify that the Trace Formatter panels are set up correctly, return to the ISPF 7.1 function "INVOKE DIALOG FUNCTION/SELECTION PANEL" and enter the REXX EXEC as shown in Figure 27.

```

----- INVOKE DIALOG FUNCTION/SELECTION          FUNCTION RC =  0

INVOKE SELECTION PANEL:
  PANEL ==>                                OPT ==>

INVOKE COMMAND:
  CMD ==> ISTTE01

  LANG ==>                                (APL or blank)

INVOKE PROGRAM/SHARED SEGMENT:
  PGM ==>                                DCSS ==>

  PARM ==>                                EXTENDED PLIST ==> NO

NEWAPPL ==> YES                                ID ==> ISTT
NEWPOOL ==> NO                                PASSLIB ==> NO

COMMAND ==>
F1=HELP    F2=SPLIT    F3=END    F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=CURSOR

```

Figure 27. Verifying the Trace Formatter Panels

The first ISPF panel you should see is shown in Figure 28 on page 102.

```

                                VTAM Internal Trace Analysis

Select one of the following. Then press Enter.

—  1. Storage Analysis
    2. Request/response unit counting
    3. VIT extraction
    4. Input complete

(C) Copyright IBM Corporation 1992. All rights reserved.
Command ==>
F1=Help      F2=Split      F3=Exit      F9=Swap      F11=Retrieve F12=Cancel
```

Figure 28. Main Menu for Selecting Trace Parameters

Press the PF1 key to verify that the appropriate help panel is displayed.

Note: It is recommended that you position the command line at the bottom of the screen using ISPF PARMs option DISPLAY and changing the 'COMMAND LINE PLACEMENT ==> ASIS' to BOTTOM to improve readability.

Customizing the ISPF Interface

If you want a customized interface to be active to select the VTAM trace analysis commands, customize the ISPF panel ISR@PRIM by adding the highlighted lines shown in Figure 29 on page 103 to create and activate option V on the ISPF/PDF Primary Option Menu as shown in Figure 30 on page 104. When this option is selected, control is passed to the ISTTE01 EXEC. This EXEC controls the ISPF panels for the trace formatter.

```

)ATTR
+ TYPE(TEXT) COLOR(GREEN) INTENS(LOW)
)BODY
%----- SAMPLE ISPF/PDF PRIMARY OPTION MENU -----+
%OPTION ==>_ZCMD
%
%
% 0 +ISPF PARMS - Specify terminal and user parameters +USERID - &ZUSER
% 1 +BROWSE - Display source data or output listings +TIME - &ZTIME
% 2 +EDIT - Create or change source data +TERMINAL - &ZTERM
% 3 +UTILITIES - Perform utility functions +PF KEYS - &ZKEYS
% 4 +FOREGROUND - Invoke language processors in foreground
% 5 +BATCH - Submit to batch for language processing
% 6 +COMMAND - Enter CMS command or EXEC
% 7 +DIALOG TEST - Perform dialog testing
% 8 +LM UTILITIES- Perform library administrator utility functions
% 9 +IBM PRODUCTS- Additional IBM program development products
% 10 +SCLM - Software Configuration and Library Manager
% C +CHANGES - Display summary of changes for this release
% V +VTAM - VTAM trace analysis commands
% T +TUTORIAL - Display information about ISPF/PDF
% X +EXIT - Terminate using console, log, and list defaults
%
+Enter%END+command to terminate ISPF.
%
)INIT
.HELP = ISR00003
&ZPRIM = YES /* ALWAYS A PRIMARY OPTION MENU */
&ZHTOP = ISR00003 /* TUTORIAL TABLE OF CONTENTS */
&ZHINDEX = ISR91000 /* TUTORIAL INDEX - 1ST PAGE */
&ZSCLMPRJ = &Z
VPUT (ZHTOP,ZHINDEX,ZSCLMPRJ) PROFILE
)PROC
&ZQ = &Z
IF (&ZCMD = ' ')
&ZQ = TRUNC(&ZCMD, '.')
IF (&ZQ = ' ')
.MSG = ISRU000
&ZSEL = TRANS( &ZQ
0,'PANEL(ISPOPTA)'
1,'PGM(ISRBRO) PARM(ISRBRO01)'
2,'PGM(ISREDIT) PARM(P,ISREDM01)'
3,'PANEL(ISRUTIL)'
4,'PANEL(ISRFPA)'
5,'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
6,'PGM(ISRPTC)'
7,'PGM(ISPYXDR) PARM(ISR) NOCHECK'
8,'PANEL(ISRLPRIM)'
9,'PANEL(ISRDIIS)'
10,'PGM(ISRSCLM) NOCHECK'
C,'PGM(ISPTUTOR) PARM(ISR00005)'
V,'CMD(%ISTTE01) NEWAPPL(ISTT)'
T,'PGM(ISPTUTOR) PARM(ISR00000)'
, , ,
X,'EXIT'
*, '?' )
&ZTRAIL = .TRAIL
)END

```

Figure 29. Sample ISPF Panel ISR@PRIM Customization

```
----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION ==>

0 ISPF PARS - Specify terminal and user parameters   USERID - USERID
1 BROWSE    - Display source data or output listings TIME   - 9:29
2 EDIT      - Create or change source data          TERMINAL - 3278
3 UTILITIES - Perform utility functions             PF KEYS  - 12
4 FOREGROUND - Invoke language processors in foreground
5 BATCH     - Submit to batch for language processing
6 COMMAND   - Enter CMS command or EXEC
7 DIALOG TEST - Perform dialog testing
8 LM UTILITIES- Perform library administrator utility functions
9 IBM PRODUCTS- Additional IBM program development products
10 SCLM     - Software Configuration and Library Manager
C CHANGES  - Display summary of changes for this release
V VTAM      - VTAM trace analysis commands
T TUTORIAL  - Display information about ISPF/PDF
X EXIT      - Terminate using console, log, and list defaults

Enter END command to terminate ISPF.
```

Figure 30. Addition of Option V on the ISPF/PDF Primary Option Menu

First Failure Support Technology (FFST)

First Failure Support Technology helps you diagnose software problems by capturing information about a potential problem when it occurs.

FFST is an IBM licensed program product and it is installed using a separate program tape and program directory. See the *FFST/VM Program Directory* for more information on installing FFST.

Installing the Online Message Facility

The online message facility is an OS/2 program that provides online information on messages. For more information, see “Using the Online Message Facility ” on page 352.

The Online Message Facility is no longer shipped as a library on the product tape. It is available on the *IBM Networking Softcopy Collection Kit* CD-ROM. Information on planning for, installing, and using the facility is in *Using the Online Message Facility*, which is available on that CD-ROM.

Installing the VTAM Command Set Library for IBM Command Tree/2

To install the VTAM command set library for the IBM Command Tree/2 program on a workstation, follow these steps:

1. Logon to the installation ID, 5654010A, or any other ID that has read access to the online OS/2 tools (401) disk. The command set library (ISTC420V) and the ISTLDRM2 unpacking utility reside on the 401 disk.
2. Download the VTAM command set library files to your workstation.

Use the OS/2 * RECEIVE command to download the files from the host. Enter the following commands on an OS/2 command line (where id is the emulator session from which you are downloading, and fm is the filemode):

```
RECEIVE ISTD420V.RAM id:ISTC420V AISTDAT1 fm
RECEIVE ISTLDRM2.EXE id:ISTLDRM2 AISTDAT1 fm
```

OS/2 downloads the ISTLDRM2 unpacking utility and the packed version of the VTAM command set library and stores them on your workstation. For more information on the RECEIVE command, enter HELP "RECEIVE" on the OS/2 command line.

For example, if you are downloading from your A emulator session, the 401 disk is accessed as filemode G, and you want to store the VTAM V4R2 for VM/ESA command set library for IBM Command Tree/2 on your D: drive in directory IBMFKB\RUNTIME\ISTC420V, enter:

```
RECEIVE D:\IBMFKB\RUNTIME\ISTC420V\ISTC420V.RAM A:ISTC420V AISTDAT1 G
```

IBMFKB is the directory that contains the IBM Command Tree/2 program files. It is suggested (but not required) that you install each command set library in a separate subdirectory of IBMFKB\RUNTIME. For example, you could store the VTAM V4R2 for VM/ESA command set library in IBMFKB\RUNTIME\ISTC420V, and a user-written command set library in IBMFKB\RUNTIME\MYCMDS.

3. Unpack the VTAM command set library files.

After the packed command set library is on your workstation, you need to unpack the files before you can build VTAM commands. To unpack the files, go to an OS/2 command line and be sure the prompt is set for the directory to which you downloaded the command set library (for example, D:\IBMFKB\RUNTIME\ISTC420V). Now enter:

```
ISTLDRM2 ISTD420V.RAM /D
```

This unpacks the files and stores them in the current directory. The ISTLDRM2 utility does not erase the packed command set library file. For backup purposes, it is recommended that you keep a copy of this packed command set library file on the workstation or on the host.

For more information on the IBM Command Tree/2 program, including how to create user profiles to start the program with the VTAM command set, see *Using IBM Command Tree/2*. This book is also available on the *IBM Networking Softcopy Collection Kit* CD-ROM.

Chapter 4. Installing VTAM under VSE

About This Chapter

This chapter describes the following steps necessary to install VTAM under the VSE operating system:

1. Calculating VTAM virtual storage requirements
2. Preparing your operating system
3. Loading VTAM
4. Starting VTAM
5. Keeping an application program active if VTAM abends.

Calculating VTAM Virtual Storage Requirements

VTAM can reside in either a shared or a private partition. If you choose a private partition, you must allocate this partition in a private address space where a large amount of allocated storage is available for VTAM to use. A small portion of VTAM code resides in these other storage spaces:

Shared virtual area (SVA)

Some VTAM code is executed in the SVA, which is built during initial program load.

System GETVIS area (SGA)

Because VTAM resides in a private partition, it must use the SGA to communicate with user tasks in other partitions.

Figure 31 on page 108 shows a sample storage layout.

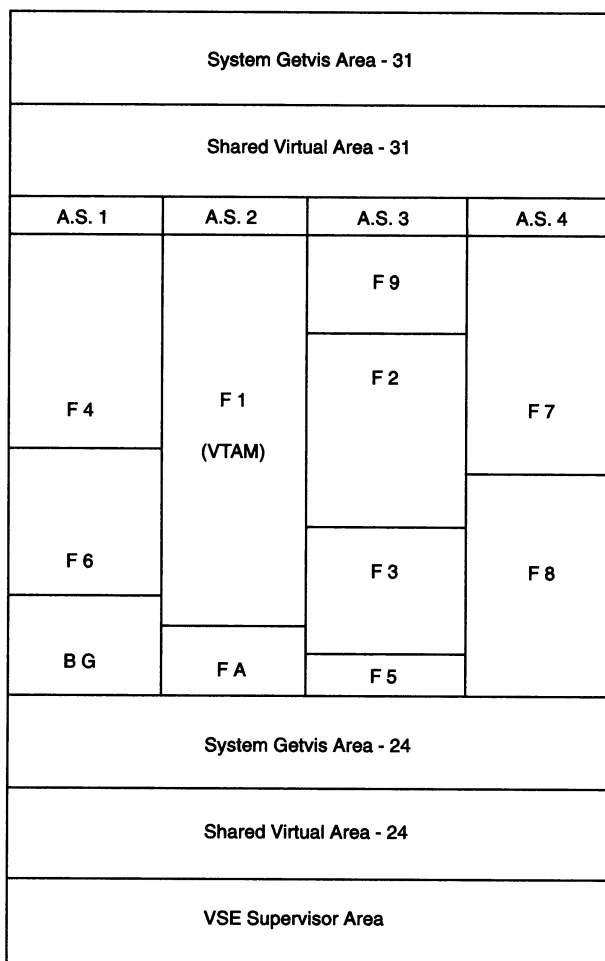


Figure 31. VSE Storage Layout

For information on defining VTAM virtual storage requirements to VSE, refer to the *Estimating Storage for VTAM* diskette.

Preparing Your Operating System

To prepare VSE to support VTAM, you will need to do the following:

- Define files for VTAM
- Define VTAM to VSE.

Defining Files for VTAM

This section describes the following:

- VTAM-related files
- Checkpointing APPN data files
- Configuration restart VSAM files
- NCP-related files
- VTAM trace files.

VTAM-Related Files

VTAM can use any of the following member types:

VTAM definition library members

These members contain VTAM definition statements, start option lists, and configuration lists. They have the suffix B in their names. VTMSNA.B is an example.

VTAM macro library members

These members contain VTAM macroinstruction definitions. They have the suffix A in their names. OPNDST.A is an example.

VTAM phase library members

These members contain VTAM phases, tables, and exit routines. They have the suffix PHASE in their names. ISTCPC00.PHASE is an example.

VTAM object library members

These members contain VTAM object-code modules that have not been link-edited. They have the suffix OBJ in their names. ISTINCAL.OBJ is an example.

For more information on the library structure in VSE, see the *VSE/Advanced Functions System Management Guide*.

Checkpointing APPN Data Files

The following file names are used when VTAM is defined as a network node or interchange node and are required for the APPN directory checkpointing and topology database checkpointing functions:

- DSDBCTL
- DSDB1
- DSDB2
- TRSDB.

DSDB1 and **DSDB2** hold APPN directory information. This information is used to initialize the directory database when VTAM is restarted. Directory database information is stored alternately between DSDB1 and DSDB2. The directory database information is written to one of the data files whenever a MODIFY CHKPT TYPE=ALL or TYPE=DIR, HALT, or HALT QUICK command is issued.

Not all of the resources from the directory database are written to the data files when there is a checkpoint. The resources that are written to the data files are those that:

- Have been the target of a search
- Have a dynamic entry type that is not registered
- Have been updated within a period of time specified by the DIRTIME start option.

The resources that are registered to the database at start up through resource registration and definition are not included in the checkpointed information.

DSDBCTL contains the current status of DSDB1 and DSDB2. It is read by VTAM during initialization to determine whether DSDB1 or DSDB2 will be used to load the APPN directory database.

TRSDB is required for checkpointing the network topology database. The information in this data file is used to initialize the network topology database whenever VTAM is restarted. The network topology database is written to this file whenever a MODIFY CHKPT TYPE=TOPO or TYPE=ALL, HALT, or HALT QUICK command is issued.

These data files should be created prior to VTAM initialization. Sample statements needed to define these files are included in Figure 32.

```
* $$ JOB JNM=VSAMAPPN,CLASS=0,DISP=D
* $$ LST CLASS=A
// EXEC PGM=IDCAMS,SIZE=AUTO

/*****/
/*                                     */
/* REALLOCATE THE VSAM CLUSTERS USED FOR VTAM APPN CHECKPOINT. */
/* DSDBCTL IS DELETE DEFINED, AND THEN DSDB1 IS DELETED/DEFINED */
/* AND DSDB2 AND TRSDB ARE MODELED AFTER IT. THIS */
/* SIMPLIFIES MAKING DEFINITION CHANGES LATER ON. */
/*                                     */
/*****/

/* MIGRATION DELETE ANY OLD COPIES OF LAN CHECK POINT FILES */
DELETE -
    (VSAM.DSDBCTL) PURGE CLUSTER

DEFINE -
    CLUSTER (NAME (VSAM.DSDBCTL) -
        RECORDS (1,1) -
        VOLUMES (SYSWKA) -
        FREESPACE (20,10) -
        NONINDEXED -
        OWNER(FVTEST) -
        RECORDSIZE (20,20) -
        REUSE -
        SHAREOPTIONS (2,3)) -
    CATALOG (VSAM.MASTER.CATALOG)

DELETE -
    (VSAM.DSDB1) PURGE CLUSTER

DEFINE -
    CLUSTER (NAME (VSAM.DSDB1) -
        RECORDS (1,1) -
        VOLUMES (SYSWKA) -
        FREESPACE (20,10) -
        NONINDEXED -
        OWNER(FVTEST) -
        RECORDSIZE (1000,1000) -
        REUSE -
        SHAREOPTIONS (2,3)) -
    CATALOG (VSAM.MASTER.CATALOG)

DELETE -
    (VSAM.DSDB2) CLUSTER PURGE
DEFINE -
    CLUSTER (NAME (VSAM.DSDB2) -
        MODEL (VSAM.DSDB1) ) -
    CATALOG (VSAM.MASTER.CATALOG)
```

Figure 32 (Part 1 of 2). Sample Definition of APPN Checkpointing Data Files

```

DELETE -
(VSAM.TRSDB) CLUSTER PURGE
DEFINE -
  CLUSTER (NAME (VSAM.TRSDB) -
            MODEL (VSAM.DSDB1) ) -
  CATALOG (VSAM.MASTER.CATALOG)
/*
/&
* $$ E0J

```

Figure 32 (Part 2 of 2). Sample Definition of APPN Checkpointing Data Files

If you do not use the sample statements provided, do the following to prepare the APPN checkpointing data files:

- Specify VSAM files for file names DSDB1, DSDB2, DSDBCTL, and TRSDB. The file names must not change but the file identifier can be changed.
- Specify the RECORDSIZE for DSDB1, DSDB2, and TRSDB as RECORDSIZE(1000,1000).
- Specify the RECORDSIZE for DSDBCTL as RECORDSIZE(20,20).

Note: We recommend that you not modify any of these data files. We also recommend that you not copy the directory data files and attempt to use them during initialization of a different node. VTAM fails the initial load of the network topology database if another node's checkpointed data file is used or if the SSCPNAME operand is changed between the two IPLs. If the initial load fails, VTAM can acquire the information dynamically through topology database updates (TDUs).

Configuration Restart VSAM Files

If you want to use VTAM's configuration restart facility, you must define configuration restart VSAM files.

To set up VSAM files for the major nodes that you are using with configuration restart, do the following:

1. Define a catalog entry to allocate space for the VSAM file. The following example shows the access method services (AMS) statements that define a configuration restart file:

```

DEFINE -
  CLUSTER(NAME(ARNCKPT) -
            VOLUMES(111111) -
            KEYS(4 0) -
            RECORDSIZE(24 136)) -
  DATA(NAME(ARNCKPT.DATA) -
         TRACKS(1)) -
  INDEX(NAME(ARNCKPT.INDEX) -
         TRACKS(1))

```

2. Code the INDEX operand on the DEFINE command, or let it default. See the sample DEFINE command in the preceding example. The file must be indexed.
3. Code KEYS (4 0). A key length of 4 bytes and an offset of 0 bytes are required.
4. Code RECORDSIZE (24 136). The average record size must be 24 bytes, and the maximum record size must be 136 bytes.

5. Make sure that the number of records in the file equals the number of minor nodes defined in the major node. (For a switched major node, include each PATH definition statement when choosing the number of records.) Therefore, the primary allocation should be the number of minor nodes in the major node, and the secondary allocation should be about one-tenth as large.
6. Make sure that the name of the configuration restart file in the DEFINE statement matches the name on the CONFGDS operand of the NCP PCCU definition statement or the VBUILD or LBUILD definition statements.

When you change a major node in the source statement library, delete and redefine the associated configuration restart file. A different file is associated with each major node.

NCP-Related Files

You may need to define the following files for your communication controller:

- A load file that contains the NCP in card-image format or the PHASE file that contains the NCP load module
- A test routine file that contains the initial test routine (which applies only to the IBM 3705 Communications Controller) of the NCP loader (one file for each host VTAM)
- A dump file to receive the NCP dump output.

NCP Load: VTAM can use either a phase or sequential data files as input to the NCP loader.

During stage 3 of the generation of an NCP, the NCP load modules are automatically link-edited into a selected sublibrary. Unless you supply the name of a different sublibrary, PRD2.CONFIG is used. The phase name is the same as the name on the NEWNAME operand of the BUILD definition statement.

When the VTAM utility that loads NCPs is invoked to load this NCP, it initially searches for a copy of the NCP in the sublibrary that you refer to in the VTAM procedure. If the phase is found in PRD2.CONFIG (or in a different sublibrary, if specified) and if enough storage has been allocated to the VTAM partition through the ALLOC statement to hold the phase, the phase is used. If there is not enough storage, VTAM looks for a copy of the NCP on a sequential data file.

If you have storage considerations, you should create sequential data files of your NCPs. To do this, punch the NCP phase to the sequential data file whose file name matches the name on the NEWNAME operand. You must provide JCL to punch the relocated NCP load image to a sequential file. SYSPCH can be assigned to a disk extent. The VSE interactive interface (IUA) provides a skeleton in the VSE interactive computing and control facility (ICCF) library that provides the JCL for this activity.

To use the VTAM utility for loading an NCP, you must add to your VTAM partition storage allocation an amount equal to the size of the largest NCP load module that you expect to process. If you do not know its size, use the directory-list output (the output from an NCP generation) to determine the correct value. You should define the NCP load file as a backup in case NCP generation changes or reallocation of operating system storage (GETVIS area) prevents use of this loading method.

VTAM dynamically assigns SYS000 to the address of the communication controller being loaded (specified on the CUADDR operand on the PCCU definition statement in the generation or by the U operand on a VARY ACT operator command).

VTAM uses some of the SSP utilities to load an NCP. The LIBDEF statement in the VTAM start procedure should have the file name of the sublibrary that contains the phases for the utilities.

The file name in the DLBL statement must be the same as the name on the NEWNAME operand of the BUILD definition statement. The symbolic device name in the EXTENT and ASSGN statements must be the same as the name on the NCPLUB operand of the PCCU definition statement.

Initial Test Routine (IBM 3705 Only): You can load the initial test routine, which is a diagnostic routine, into a channel-attached IBM 3705 Communications Controller and have it execute before the NCP is loaded. DLBL and EXTENT statements define the file that contains this initial test routine. The file name on the DLBL statement must be DIAGFLE. The LUB must be SYS008. The initial test routine is distributed with the NCP installation tape.

NCP Dump: The NCP dump utility writes the contents of the communication controller storage into a direct access device in 512-byte records. The file name in the DLBL statement must be NCPDUMP. The symbolic device name in the ASSGN statement must be the name on the DUMPDS operand of the PCCU definition statement. This symbolic device name must be omitted if more than one NCP is to be dumped.

The dump file must accommodate a dump of the entire communication controller storage. The size of communication controller storage depends on the model number.

VTAM dump processing fails if the NCP modules that are loaded to process the dump are either not in the proper sublibrary or are in a different sublibrary during dump processing.

VTAM Trace Files

The trace information is written sequentially onto a disk or tape file. If the file is on a disk, it must be defined in DLBL and EXTENT statements.

For a disk trace file, the names TRFILE and SYS001 are mandatory. TRFILE must be the first operand of the DLBL statement.

For a tape trace file, assign SYS001 to an unlabeled tape.

To use TPRINT to print a trace in batch, include SYS004.

When the disk trace file is full, the file is overwritten, with the newest trace records overlaying the oldest trace records. Subsequently, the trace-print facility can be used to write the trace information onto SYSLST.

The trace records are written to TRFILE in 2048-byte blocks. The number of tracks of direct-access storage to be allocated depends on the number of 2048-byte records that you want to save. *VTAM Diagnosis* contains additional information about the traces available in VTAM.

Defining VTAM to VSE

VSE system generation statements include VTAM support. You can change the values coded on some system generation statements, such as the SUPVR macroinstruction.

SUPVR Macroinstruction

A VSE system must have at least two partitions: one for VTAM and one for an application program. If additional partitions are needed (for other application programs or for programs unrelated to VTAM), change the NPARTS operand to include them.

The partition containing VTAM should have higher priority than partitions containing VTAM application programs. Otherwise, the application programs may not be able to communicate with VTAM (for example, a request to open an access method control block could fail).

Coding an IPL Procedure with Support for VTAM (Optional)

An initial program load (IPL) procedure loads VSE and defines to it the devices that are to be attached to it and the sizes of the partitions that it is to create. Some of the devices defined in the IPL procedure are used by VTAM. Following are examples of these devices:

- Local non-SNA 3270
- Local SNA 3270
- 3705, 3725, 3720, and 3745 communication controllers
- Token-ring ports
- Channel-to-channel and communication adapter lines.

The VSE interactive interface provides a table that maintains the VSE device definitions that the user may alter by using interactive dialogs. The new definitions are effective with the next IPL of VSE.

Following is an example of IPL statements:

```
01F,$$A$SUPI,VSIZE=120M,VI0=512K,VP00L=128K,LOG
ADD 01F,3277
ADD 00C,2540R
ADD 00D,2540P
ADD 00E,1403
ADD 030:04F,3705,10      ICA LINE
ADD 050:057,3791L        LOCAL SNA 3174
ADD 040,3725,01          CHANNEL
ADD 181:183,3420T9,D0
ADD 281:283,3480
ADD 370:37F,3380
ADD 440:44F,3277
ADD 720:72F,3277
ADD FEC,3505             POWER DUMMY READER, DO NOT DELETE
ADD FED,2520B2            POWER DUMMY PUNCH, DO NOT DELETE
ADD FEE,PRT1              POWER DUMMY PRINTER, DO NOT DELETE
ADD FEF,PRT1              POWER DUMMY PRINTER, DO NOT DELETE
ADD FFA,3505              ICCF INTERNAL READER, DO NOT DELETE
ADD FFC,3505              ICCF DUMMY READER, DO NOT DELETE
ADD FFD,2520B2            ICCF DUMMY PUNCH, DO NOT DELETE
ADD FFE,PRT1              ICCF DUMMY PRINTER, DO NOT DELETE
ADD FFF,CONS              DUMMY CONSOLE, DO NOT DELETE
```

```

SET ZONE=WEST/00/00
DEF SYSCAT=DOSRES
DEF SYSREC=SYSWK1
SYS JA=YES
SYS BUFSIZE=1500
SYS NPARTS=44
SYS SEC=NO
SYS PASIZE=30M
SYS SPSIZE=0K
SYS BUFLD=YES
DPD VOLID=DOSRES,CYL=211,NCYL=15,TYPE=N,DSF=N
DPD VOLID=SYSWK1,CYL=446,NCYL=15,TYPE=N,DSF=N
DPD VOLID=DOSRES,CYL=414,NCYL=14,TYPE=N,DSF=N
DPD VOLID=DOSRES,CYL=428,TYPE=N,DSF=N
DLA NAME=AREA1,VOLID=DOSRES,CYL=64,NCYL=3,DSF=N
SVA SDL=300,GETVIS=256K,PSIZE=(256K,1800K)

```

Defining VTAM Devices: Insert an ADD command in the IPL procedure for each channel-attached device and each communication-adapter attached line. ADD commands relate device types to specific channel device names. For more information about the device type codes for specific devices, see the description of the ADD command in *VSE/ESA System Control Statements*.

Defining Sizes of VTAM and Application Program Partitions: During VSE IPL, the VTAM SVA storage requirements must be defined using the SVA command via the PSIZE and GETVIS operands. After VSE IPL, another procedure 0JCLxxx (for example \$0JCL370) is automatically invoked. Virtual storage allocation for each partition is contained in this procedure (including VTAM's virtual storage allocation). The sizes of VSE system partitions are defined during system generation by the ALLOC operator command from the background procedure. Virtual storage must be allocated for the partition in which VTAM runs. Real storage should be defined on the SETPFIX statement. For information on calculating VTAM's virtual storage allocation, refer to the *Estimating Storage for VTAM* diskette.

Defining Data Space Virtual Storage: Define data space virtual storage as follows:

1. Define the total amount of virtual storage that can be allocated for data spaces. This is done using the DSIZE operand on the VSE system control statement, SYSDEF.

When determining a value for the DSIZE operand, remember that VTAM always requires a 1 Mb data space for initialization. In addition, for each partition in which a VTAM application program is running, VTAM requires an additional 1 Mb data space. For example, if you have CICS/ESA running in a partition, VTAM requires a 1 Mb data space to support CICS/ESA sessions, plus the additional 1 Mb data space for VTAM initialization, for a total VTAM data space requirement of 2 Mb. If you have a VTAM user-written application program running in another partition, VTAM will require another 1 Mb data space to support that application program's sessions, now bringing the total VTAM data space requirement to 3 Mb.

2. Define the maximum size of each data space for VTAM application programs.

Data spaces required by VTAM expand in 1 Mb increments, depending on data space storage needed to support an application program, up to whatever you

define as the maximum size. To define this maximum size for VTAM application programs, use one of the following:

- The DFSIZE operand on the VSE SYSDEF system control statement
- The DSPACE operand on the EXEC JCL statement for a VTAM application program.

The value you specify on either operand should be a multiple of 1 Mb.

The DFSIZE operand is a global system value and is used for each application program that uses data space storage. You can override this value for a particular VTAM application program by specifying a different value on the DSPACE operand for that VTAM application program. If you do not code the DSPACE operand on the EXEC JCL statement for a VTAM application program, the value you specify on the DFSIZE operand on the VSE SYSDEF system control statement is used, unless that value is less than 1 Mb. If the value you specify on the DFSIZE operand is less than 1 Mb, VTAM will use a maximum data space size of 1 Mb.

Note: The value you specify on the DSPACE operand is passed to VTAM so that VTAM knows the maximum size to allow for data spaces it creates to support the application program. The value you specify on the DSPACE operand does not affect the maximum size for data spaces created in the application program itself (defined on the MAX operand on the DPSERV macroinstruction).

Data spaces are divided into .25 Mb blocks. Each block is used for request/response units (RUs) whose sizes fall into a certain range. If an application program sends many different RU sizes, you might need to define a maximum data space size greater than 1 Mb. For example, if an application program sends RU sizes ranging from X'80' to X'1000', you might need to define a maximum data space size of 2 Mb.

VTAM uses only the data space storage that it needs. If you define a maximum data space size of 3 Mb, but RU sizes do not vary much, expansion of the data space is not required. The data space size remains at 1 Mb. You do not affect the available virtual storage in your system if you define a maximum data space size larger than what you actually need.

Although data space storage is most often associated with persistent sessions, non-persistent sessions use data spaces too. This is especially true when data has been sent, but a non-persistent application program is not ready to receive it.

With non-persistent sessions, RU sizes are likely never to require a maximum data space size greater than 3 Mb; most will use only 1 Mb or 2 Mb. With persistent sessions, however, data space sizes are more likely to expand, especially if large pieces of data are sent and an application program is not able to receive the data. By setting a limit on the size to which the data spaces can expand, you prevent data spaces from expanding to the point that they use all available virtual storage.

Loading VTAM

The *VTAM Program Directory* that is shipped with VTAM describes the procedure for installing VTAM from the distribution tape, including suggested coding statements. Installing VTAM under VSE/ESA consists of restoring the VTAM production libraries and updating the VSE system history file using the VSE maintain system history program (MSHP). The *VTAM Program Directory* also describes the contents of the tape and space allocations required on system files. The tape contains the VTAM load modules, phases, and source code needed to make VTAM part of VSE.

Before you can use VTAM, it must be installed into the VSE operating system during system generation. Once VTAM is installed, you do not have to perform a VSE system generation when you are migrating to a new release of VTAM. VTAM support is a standard feature of VSE.

VSE/ESA is installed as a package or group of software products supported by VSE/ESA. Some of these software products are installed as part of VSE, and some are optional. VTAM is a part of VSE/ESA and is automatically installed with the package. When VTAM is installed as part of VSE/ESA, the installation procedure selects PRD1.BASE as the library.sublibrary for VTAM.

For information about planning for the installation of VSE/ESA and about installing VSE/ESA, refer to *VSE/ESA Installation*.

Starting VTAM

You can issue an EXEC statement to begin the execution of VTAM one of the following ways:

- You can create a procedure to start VTAM (called a start procedure) for EXEC.
- You can submit a job stream from the VSE/POWER spool containing the JCL to start VTAM.
- You can use VSE facilities for conditional JCL to dynamically modify the start procedure for VTAM.

The manner in which you issue EXEC, enter the JCL, and invoke or modify a start procedure is not unique for VTAM; these actions are done as described in *VSE/ESA Operation*.

Following is an example of job control statements for a typical start procedure.

```
// JOB ACFVTAM
// ASSGN SYS000,UA                                1
// LIBDEF *,SEARCH=(PRD2.COMM,PRD2.CONFIG,PRD2.BASE) 2
*
// ASSGN SYSLST,cua                                3,4
*
// DLBL DIAGFLE,'IFUNCPD,IFUNCPE'
// EXTENT SYS008
// ASSGN SYS008,cua
*
// DLBL NCP001,'NCP001.LOAD.FILE'
// EXTENT SYS007
// ASSGN SYS007,cua
```

```

*
// DLBL NCPDUMP,'NCPDUMP.FILE',,DA
// EXTENT ,,,190,95
// ASSGN SYS005,DISK,VOL=VOLIDS,SHR
*
* OPTIONAL TRACE FILE
// DLBL TRFILE,'VTAM.TRACE.FILE'
// EXTENT SYS001
// ASSGN SYS001,cua
*
* OPTIONAL CONFIGURATION RESTART FILE
// DLBL ANYNAME,,,VSAM
// EXTENT SYSCAT,VSAM01
*
* OPTIONAL CONFIGURATION RESTART FILE
// DLBL NCP001C,,,VSAM
// EXTENT SYSCAT,VSAM01
*
* OPTIONAL NODELST FILE
// DLBL NODELST,,,VSAM
// EXTENT SYSCAT,VSAM01
// SETPFIX LIMIT=(500K,600K)
// EXEC ISTINCVT,PARM='CUSTNO=C555-555-5555,VTAMPW=4394-0080-1454-3481-X 5
                        8008,LIST=xx',DSPACE=2M
*
* OPTIONAL DUMP OPTIONS
// STDOPT DUMP=YES
// OPTION DUMP
*

```

6

The following notes are keyed to the numbers shown in the right margin of the preceding example.

Notes:

1. Temporarily unassigned so VTAM can use the symbolic name SYS000 for the communication controller when loading an NCP. VTAM dynamically assigns SYS000 to the communication controller channel device name during LOAD and DUMP. SYS000 is also used when connecting or disconnecting channel-attached SNA cluster controllers.
2. VTAM and NCP phases, macros, and modules along with user defined VTAM and NCP definitions reside in a group of library.sublibrary that the system searches from left to right when loading is required. In this example, the system searches the NCP library, the user definition library, and the VTAM library, respectively. Each one of these library.sublibraries contains phases, modules, and macros. The system library.sublibrary default is IJSYSRS.SYSLIB, and it is searched last (by default).
3. For the trace printout, SYSLST must be assigned. When the trace file is full, the trace print subtask of VTAM can be used to write the trace information onto SYSLST. With this method, the trace is printed when VTAM is terminated or if segmentation is used in the VTAM job. An alternate method to print the trace without terminating VTAM is to run TPRINT in a batch job. For details on how to run TPRINT, refer to *VTAM Diagnosis*.
4. If a controller channel I/O error occurs while you are loading NCP, the loader produces a trace table containing information on the channel programs exe-

cuted by the loader. To obtain this trace table, see the *NCP, SSP, and EP Diagnosis Guide*.

5. A customer number (CUSTNO) and VTAM password (VTAMPW) must be included in the order shown as the first parameters on the EXEC statement. By default, the CUSTNO and VTAMPW provided can be used to enable the VTAM Client/Server package. However, IBM recommends use of your own CUSTNO and VTAMPW. Edit the start procedure and substitute your CUSTNO and VTAMPW for C555-555-5555 and 4394-0080-1454-3481-8008 respectively. To use VTAM MultiDomain or VTAM InterEnterprise, you must edit the start procedure to include your CUSTNO and VTAMPW. For more information on the different VTAM packages, see the *VTAM Overview for VM/ESA and VSE/ESA*.

The LIST parameter can be coded on the EXEC statement (after CUSTNO and VTAMPW) to start VTAM without prompting for a start list. Errors in start list xx will result in the default start list, ATCSTR00, being used.

Only the CUSTNO, VTAMPW, and LIST start parameters are allowed on the EXEC statement. If any other parameter is specified, the operator will be prompted for the start list.

6. When VTAM abends, multiple dumps (IDUMPs) are taken:
 - Dump of the shared virtual area
 - Dump of the VTAM partition, VSE supervisor storage, and used portions of the system GETVIS area
 - Dump of each data space owned by VTAM.

All dumps are necessary for complete documentation of the abend. To receive a complete dump of the VTAM partition, VSE supervisor storage, and used portions of the system GETVIS area, specify the DUMP option on the VTAM start job. If you specify the PARTDUMP option instead, VSE supervisor storage is not dumped. Without this part of the dump, key storage areas are not available.

Keeping an Application Program Active If VTAM Abends

Certain VTAM modules are loaded into SVA during the IPL procedure. These modules will keep an application program active if VTAM abends.

The load procedure occurs in \$OJCLxxx as provided by the operating system. The procedure will load the VTAM SVA modules using the \$SVAVTM load list. VTAM initialization will fail if the necessary modules have not been loaded into SVA.

Do not use USERBG.PROC to load the VTAM SVA modules. USERBG.PROC is controlled by VSE/POWER, which is activated after VTAM initializes. Therefore, if you use USERBG.PROC, the modules are loaded after VTAM initializes, which is not allowed.

Part 2. Getting Started with VTAM

Chapter 5. Getting Started with VTAM - Overview	125
Implementing a VTAM Network	125
Using Start Options and Configuration Lists	127
Start Options	127
Configuration Lists	134
Identifying Resources to VTAM	136
Coding Concepts	136
Defining Application Programs	138
Attaching NCP to VTAM	138
Defining Peripheral Nodes and Logical Units	139
Attaching Peripheral Nodes to VTAM	142
Attaching Peripheral Nodes to NCP	142
Defining Switched Resources Dynamically	143
Attaching Other Domains to VTAM	145
Sharing NCP Peripheral Nodes (CMC Configurations)	145
Dynamic Configuration of Channel-Attached Devices	147
Dynamic Reconfiguration and Change of Operands	149
Identifying Routes and Establishing Sessions	150
Network Routing for Subarea Nodes	151
Establishing and Controlling SNA Sessions	155
Operating VTAM	159
Verifying a VTAM Network	160
Chapter 6. Getting Started with VTAM - Detail	161
Application Programs	161
Naming an Application Program	161
Authorizing Application Facilities	162
Data Compression	163
Security Facilities	168
Logon Mode Parameters	173
High Availability Using Extended Recovery Facility	174
Cryptography in an XRF Environment	175
Using User Variables (USERVAR)	175
Generic Resources	182
LU 6.2 in an XRF Environment	193
Persistent LU-LU Sessions	193
VTAM Common Network Services	195
Cross-Memory Application Program Interface (API)	195
Allocating Private Storage	195
Communicating with Start-Stop Devices	196
LU 6.2 Application Programs	196
LU 6.2 Security	199
LU 6.2 Sync Point Services	201
Selective Termination of Idle LU 6.2 Sessions	201
Attaching NCP to VTAM	202
NCP Definition Statements for VTAM	202
NCP Definition Statements for NCP	207
Channel-Attached NCP	207
Using Parallel Transmission Groups between VTAM and a Channel-Attached NCP	209

Link-Attached NCP	212
Defining Dependent LUs Dynamically	216
Independent LUs	218
Characteristics of Independent LUs	218
Defining Independent LUs	220
Multiple Connections between a Type 2.1 Node and a Subarea Node	224
Dynamic Selection of Session Connections	225
Connecting Independent LUs through NCPs	226
Authorized Transmission Priority for LEN Connections	227
Restrictions on Using Independent LUs	227
Attaching Peripheral Nodes to VTAM	228
Channel Connections	228
External Communication Adapter (XCA) Connections	231
Integrated Communication Adapter (ICA) Connections	234
Attaching Peripheral Nodes to NCP	249
Defining Type 2.1 Peripheral Nodes	249
Non-Native Network Type 2.1 Connections	250
NCP Connections	250
Dynamically Defining Switched Resources	260
Attaching Other Domains to VTAM	262
VTAM-to-VTAM Channel Connections	262
VTAM-to-VTAM NCP Connections	267
VTAM-to-VTAM XCA Connections	274
VTAM-to-VTAM ICA Connections	278
Sharing NCP Peripheral Nodes (CMC Configurations)	284
Channel-Attached NCPs in a CMC Configuration	284
Link-Attached NCPs in a CMC Configuration	286
Controlling Resource Ownership in a CMC Configuration	286
Problem Determination in a CMC Configuration	287
Dynamic Configuration of Channel-Attached Devices	287
Installation and Preparation	287
Defining Your Configuration	288
Building Resource Definitions	288
Using the Default Naming Convention	289
Customizing the Command Lists	290
Dynamic Reconfiguration and Change of Operands	291
Dynamic Reconfiguration and Dynamic Change Requirements	291
Using the VARY ACT,UPDATE Technique	292
Using the VARY DRDS Technique	295
Using the MODIFY DR Technique	297
Dynamic Reconfiguration of Independent LUs	297
Dynamic Reconfiguration of Frame Relay Resources	298
Implicit Dynamic Reconfiguration	300
Dynamic Reconfiguration Examples	301
Line Backup through Dynamic Reconfiguration	304
Network Routing for Subarea Nodes	306
How VTAM Handles Network and Subarea Addressing	306
Virtual Route Pacing	310
Parallel Sessions Using Parallel Transmission Groups	312
Dynamic Path Update	313
Establishing and Controlling SNA Sessions	315
National Language Support for End-User USS Messages and Commands	316
National Language Support for the Language Passed to Application Programs	317

Model Name Table	318
Associated LU Table	319
Selecting Session Parameters for the Logon Mode Table	320
Automatic Logons	323
Session Authorization and Accounting	329
Pacing and Flow Control	330
Application Program Pacing	337
Setting Pacing Values	339
Operating VTAM	341
Starting the Domain	341
Activating Resources	348
Monitoring the Domain	350
Controlling the Domain	354
Deactivating Resources	355
Halting VTAM	356
Canceling VTAM	358
Automatic Operations	358
Operating with an NCP	361
Loading an NCP	362
Activating an NCP	371
Monitoring an NCP	379
Deactivating an NCP	381
Dumping an NCP	382

Chapter 5. Getting Started with VTAM - Overview

This chapter provides an overview of VTAM implementation issues common to all VTAM networks. Chapter 6, "Getting Started with VTAM - Detail" on page 161 provides details and should be used as a reference when necessary as you read this overview. Subsequent chapters describe additional implementation issues specific to particular environments. For example, for information on VTAM implementation issues specific to an APPN network, see Chapter 10, "Implementing an APPN Network" on page 525.

Throughout this chapter, the term *file* represents the following:

MVS	Partitioned data set member
VM	File
VSE	Book.

Implementing a VTAM Network

Implementing a VTAM network involves the use of start options and an optional configuration list, identifying particular resources in the network to VTAM, identifying any paths required for network routing, establishing sessions, and operating VTAM. Below is a brief description and an example of implementing a VTAM network. Subsequent sections of this chapter provide further information on these and other topics.

Start options and configuration list:

Start options control the conditions under which VTAM runs. No matter what network configuration you are implementing, you need to code some start options. Start options are coded in files named ATCSTRxx, where xx specifies the identifier of a particular start option file.

A configuration list lets you specify which resources are activated automatically when you start VTAM. You are not required to define a configuration list, but it makes the VTAM operator's job easier by activating desired resources automatically. Configuration lists are coded in files named ATCCONxx, where xx specifies the identifier of a particular configuration list.

Application programs:

Each host application program you are running must be defined to VTAM as an application program minor node within an application program major node. A major node, such as an application program major node, is a set of minor nodes (in this case, the application programs) that can be activated and deactivated as a group.

Note: An application program may become a shadow resource if a CDRSC with the same name already exists when the major node containing the application program's definition is activated. For more information on shadow resources, see "Shadow Resources" on page 407.

Subarea nodes:

If you have one or more NCPs in your network, define them in NCP major nodes and channel-attachment major nodes.

If there are other VTAMs in your network, define connections to them in NCP major nodes, channel-attachment major nodes, or external commu-

nication adapter (XCA) major nodes. If your VTAM is to have SSCP-SSCP sessions with other VTAMs, also create a cross-domain resource manager major node and minor nodes for your VTAM and adjacent VTAMs with which your VTAM is to have such sessions.

Peripheral nodes:

Other physical devices in your network must also be defined to VTAM, whether they are directly attached to the host or to an NCP. Peripheral nodes are dynamically defined or manually defined in channel-attachment, cross-domain resource, external communication adapter, local non-SNA, local SNA, LU group, model, packet, and switched major nodes. The major node you choose depends on the characteristics of the device you are defining.

Paths for network routing:

You need to define paths for data flow between VTAM and any owned NCPs, and for data flow to and from other VTAMs and NCPs in your network over non-APPN connections. Paths are defined in PATH definition statements.

Figure 33 is a sample VTAM subarea network. The host is running MVS, VTAM, and two VTAM application programs. NCP11 is a channel-attached 3745 controller, and NCP12 is a link-attached 3725 controller. Peripheral nodes are connected using leased lines.

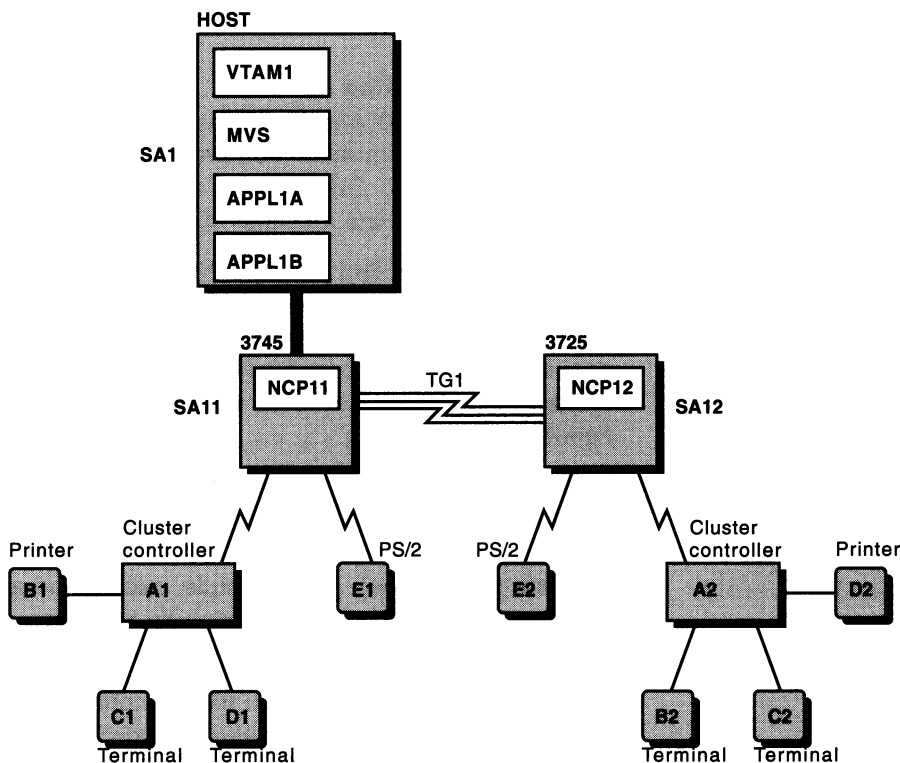


Figure 33. VTAM Network

You need to code at least the following resources to define the network in Figure 33:

- Start options describing the host (NETID, SSCPID, and SSCPNAME).
- Application program minor nodes defining APPL1A and APPL1B.

- Two NCP major nodes defining NCP11 and NCP12. These include the NCP definitions. In the NCP major node, also code GROUP, LINE, PU, and LU definition statements to define the peripheral nodes attached to the NCPs.
Note: PUs and LUs can be defined dynamically.
- PATH definition statements in each subarea to attach to the other subareas. For example, in NCP11, you would code two PATH definition statements to attach to subarea 1 (DESTSA=1) and subarea 12 (DESTSA=12).

Using Start Options and Configuration Lists

When you start VTAM, you can enter start options on the START command, or you can write one or more start lists to assist you in starting VTAM. You can also activate your resources either manually or automatically. To automatically activate some or all of your resources, you can write one or more configuration lists.

Writing start lists and configuration lists:

- Reduces the amount of operator involvement and the chance of entering incorrect information
- Enables VTAM to initialize the domain faster.

You can code multiple start options or configuration lists and use these lists according to your needs. After you specify start options and configuration lists, they remain in effect until you modify them or until VTAM is halted.

Start Options

The following topics are discussed in this section:

- Required start options
- Recommended start options
- Sources of start options
- Start option processing
- Creating start option lists
- Improving VTAM performance using start options
- Diagnosing problems using start options.

Start options provide information about the conditions under which VTAM runs. They also enable you to tailor VTAM to meet your needs each time VTAM is started. Many operands can have defaults specified as start options, thus reducing the amount of coding required. Many start options can be dynamically modified and also displayed. A complete list of start options is listed in Chapter 4, "Start Options" in the *VTAM Resource Definition Reference*.

Required Start Options

The only required start options are SSCPID, SSCPNAME, and NETID.

SSCPID: The SSCPID start option provides VTAM with a unique numeric identifier. The SSCPID value is used by some physical units to identify the VTAM with which it is in session.

If you plan to expand or incorporate a single-domain network into a larger network, be sure that the value of SSCPID is unique for each host. The SSCPID value you specify must be different from the SSCPIDs in other networks that can be in session with this host.

SSCPNAME: The SSCPNAME start option provides a unique name for VTAM. It is required for a single-domain network, but is primarily used in multiple-domain and multiple-network environments to identify a particular VTAM. The SSCPNAME value must be different from the HOSTPU start option that identifies the physical unit within VTAM.

The SSCPNAME value is also used as the CP name for a VTAM that implements APPN. Although the SSCPNAME and CP name are identical, note that the SSCP and CP are distinct resources and can be displayed individually.

NETID: The NETID start option provides VTAM with the network identifier. If you connect your VTAM to another network, the network identifiers must be unique. The network name should conform to an overall naming standard, such as the following:

- The first two characters identify the country in which the network is managed and comply with the International Standards Organization (ISO) Standard 3166.
- The next four characters identify your enterprise.
- The last two characters identify a particular network within your enterprise.

For example, an IBM network within the United States could have the network identifier *USIBMCMK*, where *US* identifies the country, *IBM* identifies the enterprise, and *MK* identifies a particular network within IBM.

Notes:

1. It is recommended that you do not use the # sign for defining network identifiers (NETIDs).
2. To assist SNA network owners in controlling the uniqueness of their network IDs, IBM has established a worldwide SNA Network Registry. Registering a network ID in the SNA Network Registry confirms its uniqueness from all other network IDs within the registry, thus minimizing future connectivity problems. To register a network ID, contact your branch office representative.

Recommended Start Options

The HOSTPU start option is recommended (but is not required) for identifying VTAM to the network.

HOSTPU: Use the HOSTPU start option to assign a user-defined name to the VTAM host physical unit. The default name for the host physical unit is ISTEPUS. Because the NetView * program uses this name to associate network information to a specific VTAM, you should make sure that this name is meaningful and unique throughout the network.

For start options that you are not required to code, VTAM uses the default values. You can override any start option value when VTAM is started or enter new start option values at the operator console.

Sources of Start Options

When VTAM is started, options can be provided from any combination of the following sources:

- IBM-supplied values internal to VTAM (default values).
- Default start option list. When you start VTAM, a list of user-defined default values (ATCSTR00) is read from the VTAM definition library. If VTAM cannot find the ATCSTR00 file, it prompts you, giving you a choice of:
 - using default values
 - using an alternate start option file (ATCSTRxx, where xx specifies the identifier of the start option file you want to use)
 - canceling the start attempt.

If you prefer not to use the ATCSTR00 file, you can avoid the prompt by coding an ATCSTR00 file that contains only comments. VTAM issues a message that the file is empty, but does not interrupt start processing.

- Supplemental start option list. The VTAM operator can enter the LIST=xx start option on the START command to name another list to supplement ATCSTR00. The supplemental list (ATCSTRxx) can override options in the default ATCSTR00 list, as well as the IBM-supplied internal default values.

VSE The VTAM operator cannot enter start options on the EXEC command. Instead, VTAM prompts the operator for start options (unless the NOPROMPT operand is coded in ATCSTR00). However, the LIST=xx parameter can be used on the EXEC statement of a start procedure to specify a start list to be used.

- Backup start option list. The LISTBKUP start option can be placed in a start list (ATCSTR00 or ATCSTRxx) to indicate the processing that occurs if a start option in the list is not valid. You should place the LISTBKUP start option first in a start list, so that it will have been processed if an error is found for another start option.
- **MVS, VM** Start options entered by the operator. The VTAM operator can enter additional start options on the START command, and also during VTAM startup if prompted (VTAM prompts for start options if the operator does not enter any start options on the START command). To prevent VTAM from prompting for start options during startup, code the NOPROMPT option in ATCSTR00.

Start options entered by the operator can override options in a specified supplemental start option list, as well as start options in the default ATCSTR00 list and the IBM-supplied internal default values.

To enter a list of options longer than the length of the console, place a comma after the last start option that will fit on the console and VTAM will continue to prompt for more start options.

- Start options reentered by the system operator to correct errors in previously processed start options.

Start Option Processing

The preceding list of start option sources is arranged from the lowest priority to the highest. VTAM processes the start options with the lowest priority first and those with the highest priority last. If conflicting specifications exist for a particular start option, the last valid specification entered always overrides any previous specification. For example, if ATCSTR00 contains CONFIG=01, but the operator enters LIST=ST and ATCSTRST contains the start option CONFIG=02, VTAM activates the configuration defined by ATCCON02.

If VTAM detects an error when reading in a start option list, it notifies the operator, who must:

- Respond to the error message by selecting the “continue processing” option and then reenter any start options that were specified incorrectly in the start option list. The operator can also enter additional options at this time.
- Specify a different start option list to use by entering xx (for the desired ATCSTRxx start option list). In effect, this cancels the start attempt and then begins start processing again with another start option list.
- Halt VTAM. In effect, this cancels the start attempt so that you can correct the start option list and try again.

After reading in a start option list, if VTAM detects an error while processing a specific start option:

- Backup processing occurs if the LISTBKUP start option was processed and the start list in error is not itself a backup start list (only one backup start list can be processed).
- The operator is provided with the same three choices identified above (that is, the choices available when VTAM encounters an error while reading in a start list) if the LISTBKUP start option was not processed.

Use the LISTBKUP start option to specify that:

- Another start list should be used in place of the start list in error (LISTBKUP=xx)
- All start option values existing before the file in error was processed should be used (LISTBKUP=DEFAULTS)
- All valid start options in the file in error will be processed, and that incorrectly specified start options should remain as previously set until VTAM reprompts for these options (LISTBKUP=PROMPT).

Creating Start Option Lists

To use a start option list, create a list and put it in a VTAMLST file named ATCSTRyy (for MVS/ESA, a partitioned data set member named ATCSTRyy in SYS1.VTAMLST). The yy value must be the same as the yy value in the LIST=yy start option entered by the operator.

When the VTAM operator enters LIST=yy as a start option, VTAM first attempts to locate ATCSTR00. If it does not exist, VTAM sends a warning message to the operator. To avoid receiving this message, create an ATCSTR00 file that contains only comments or start options that are always used for that particular VTAM.

Following is a sample start list ATCSTRyy. The scale is included to illustrate the coding restrictions (for example, the continuation character in column 72).

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
SSCPID=01, X
SSCPNAME=SSCP1A, X
NETID=NETA, X
DYNASSCP=YES, X
SSCPDYN=YES, X
HOSTSA=1
```

The operator can start VTAM with this list using LIST=yy on the start command.

One start list cannot refer to another using LIST=yy. Only the operator can enter LIST=yy to use a start list.

Notes:

1. **VM** Whenever you change start option lists, you must use the GCS ACCESS command to re-access the disk that contains the corresponding VTAMLST files before you restart VTAM.
2. **VM** Code start options for VSCS as operands on the DTIGEN macroinstruction. "VSCS Start Options" on page 633 describes how to specify start options for VSCS. The *VTAM Resource Definition Reference* describes the DTIGEN macroinstruction and its operands.

Improving VTAM Performance Using Start Options

This section discusses two ways to improve performance using start options:

- VTAM's search reduction support can be used with the SRCHRED, SRTIMER, and SRCOUNT start options
- The NCPBUFSZ start option can be used to improve performance when you load or dump a link-attached 3720 or 3745 communication controller.

Your use of start options also affects VTAM's use of host processor time and storage. Buffer pool specifications, the limit of buffer usage for session outage notification, and the number of subtasks used to perform dump, load, and restart processes impacts VTAM host resources. For more information about these start options and other factors that impact VTAM performance in your network, see Chapter 11, "Tuning VTAM for Your Environment" on page 565.

Search Reduction: When a resource is unreachable in a network, futile attempts to reach it can still occur. Excessive searching for unreachable resources can adversely affect network performance. Therefore, VTAM provides search reduction support, which limits requests for resources that have been found to be unreachable.

If search reduction support is enabled, an APPN or subarea search reduction entry is created when a resource discovery search (RDS) fails to locate a target resource. An APPN search reduction entry is saved as a directory entry in the directory services database. A subarea search reduction entry is saved as a CDRSC in the subarea resource definition table (RDT). VTAM then reduces searching for the unreachable resource in the appropriate part of the network during a designated amount of time or until a designated number of search requests have been received for the resource.

If a subarea search reduction entry exists for a destination resource and VTAM is the SSCP of the originating LU (including instances where VTAM is an interchange

node for requests entering subarea from APPN), search reduction optimizes adjacent SSCP table searching for that resource; VTAM limits attempts during the designated period to locate the unreachable resource in the subarea network.

If an APPN search reduction entry exists for a destination resource, the scope of optimization varies according to the node type of VTAM:

- If VTAM is the central directory server or network node server for the originating LU, including instances where VTAM is an interchange node for requests entering APPN from subarea, search reduction optimizes APPN searching in VTAM's subnetwork; VTAM limits attempts during the designated period to locate the unreachable resource in its APPN subnetwork. For a discussion of APPN subnetworks, see "APPN Multiple Network Connectivity" on page 545.
- If VTAM is a border node for requests received across a subnetwork boundary, search reduction optimizes APPN searching in VTAM's subnetwork; VTAM limits attempts during the designated period to locate the unreachable resource in its APPN subnetwork.
- If VTAM is a border node for the originating LU, search reduction optimizes APPN searching throughout the APPN network; VTAM limits attempts during the designated period to locate the unreachable resource in the APPN network.

To use VTAM's search reduction support, code the SRCHRED=ON start option or enter the MODIFY VTAMOPTS command with the SRCHRED=ON operand. Use the SRTIMER start option, or MODIFY VTAMOPTS with the SRTIMER operand, to designate the amount of time that VTAM should limit requests for a resource. Use the SRCOUNT start option, or MODIFY VTAMOPTS with the SRCOUNT operand, to designate the number of search requests for a resource that VTAM should limit before attempting to locate the resource again.

In determining a proper value for SRCOUNT, note that the search threshold counter (determined by SRCOUNT) can be decremented multiple times for a single end-user request, if multiple search requests are required to process the end-user request. To account for this, you can determine the maximum number of end-user requests that should be limited by a search reduction entry before verification should occur, then multiply by two.

In determining a proper value for SRTIMER or SRCOUNT, also keep in mind that the value should be large enough to limit searching for an appropriate amount of time, but small enough so that a resource that becomes available again does not remain unreachable due to waiting for SRTIMER or SRCOUNT to expire. In addition, keep in mind that the SRCLEAR=YES operand can be used on the MODIFY RESOURCE command to clear search reduction entries. That is, you can use this when a resource becomes available to clear search reduction entries, so that the specified resource can now be located.

Note that SRCLEAR=YES only clears search reduction entries, it does not turn off search reduction support. If a subsequent RDS fails to locate the target resource, VTAM creates a new search reduction entry and limits subsequent requests for the unreachable resource during the designated amount of time or until the designated number of requests have been made.

Notes:

1. The SRTIMER and SRCOUNT operands can also be specified for specific resources on the CDRSC definition statement or the GROUP definition statement in a CDRSC major node. These values can be dynamically modified using the MODIFY RESOURCE command.
2. The DISPLAY DIRECTRY command displays information about search reduction entries in the directory services database. The DISPLAY ID command displays information about search reduction entries for a particular CDRSC, as well as entries in the directory services database (if the IDTYPE=RESOURCE operand is used).

When a resource has search reduction information associated with it, these commands display the defined values for SRTIMER and SRCOUNT, and the remaining amount of time and remaining number of searches before VTAM discards the search reduction information.

3. Use DISPLAY VTAMOPTS for information on the settings of the search reduction start options.
4. When using SRTIMER, SRCOUNT, or both, along with AUTOTI for autologon sessions, an unsuccessful autologon-session search for a controlling PLU can cause the creation of a search reduction entry for it. In this situation, be aware of the following:
 - If AUTOTI=x and SRTIMER=y, where y is greater than x, when the AUTOTI timer expires, subsequent requests for the PLU are limited by the search reduction entry until SRTIMER also expires.
 - If AUTOTI=x and SRCOUNT=z, when the AUTOTI timer expires, subsequent requests for the PLU are limited by the search reduction entry for (x * z) seconds. At that time, the search reduction entry expires.

Specifying a Request Unit Size: To improve performance when you load or dump a link-attached 3720 or 3745 communication controller, you can specify the RU size to be used to transfer either the load module to or the static dump from the communication controller. The default size is 512 bytes, but values of 1024 or 2048 can be specified on the NCPBUFSZ start option to improve the performance of the load or dump operation.

VTAM determines the RU size to be used for a dynamic dump. VTAM always attempts to use the 2K transfer for a dynamic dump but uses a 256-byte transfer under any of the following conditions:

- The SSP does not support a 2K buffer for the transfer operation.
- The NCP does not support a 2K transfer operation.
- The lines between VTAM and NCP do not support the use of 2K path information units (PIUs).

Diagnosing Problems Using Start Options

The following problem diagnosis aids are available to you. You can implement these when you start VTAM.

NMVTLOG start option

Enables you to specify whether error records are written to the operating system's error data set (LOGREC). If NMVTLOG=NPDA (the default), VTAM records alerts in LOGREC only when the NetView hardware

monitor, NPDA, is not active. If NMVTLOG=ALWAYS, VTAM always records alerts in LOGREC. If NMVTLOG=NEVER, VTAM never records them.

SDLCMDRS start option

Enables you to specify whether SDLC inoperative and statistical miscellaneous data records (MDRs) are written to LOGREC. If SDLCMDRS=YES (the default), VTAM records SDLC inoperative and statistical MDRs in LOGREC.

BSCMDRS start option

Enables you to specify whether subarea BSC inoperative and statistical miscellaneous data records (MDRs) are written to LOGREC. If BSCMDRS=(STATS,INOPS), which is the default, VTAM records subarea BSC inoperative and statistical MDRs in LOGREC.

MSGMOD start option

Enables you to see the name of the module from which a message is issued. If you specify MSGMOD=YES, VTAM inserts the last five characters of the module name that issued the message, following the VTAM message identifier (for example, ISTnnnl). However, using MSGMOD causes the last five characters of the message to be truncated if the message exceeds the maximum console message length allowed by the operating system.

IOINT start option

Enables you to request that VTAM inform the operator when a VTAM request is not answered after a specified interval of time. After the period of time specified by IOINT, VTAM issues a message to the operator. If VTAM issues the message multiple times for the same request or several different requests, initiate the appropriate diagnostic actions to resolve the problem.

TRACE start option

Enables you to collect information such as data that flows through VTAM's buffers, data transmitted on NCP links, and VTAM internal trace information. For more information on the trace facility, see "Activating Network Traces" in *VTAM Diagnosis*.

For a list of the start options and for descriptions of the operator commands that you can use to start VTAM trace facilities, see *VTAM Operation*. For descriptions of the start options, see Chapter 4, "Start Options" in the *VTAM Resource Definition Reference*.

Configuration Lists

A configuration list specifies the resources that are to be activated when you start VTAM. Place the member names for the desired resources into an ATCCONxx member in the VTAM definition library, where xx is any two alphanumeric characters. The value xx can then be used on the CONFIG operand of the VTAM START command, or on the CONFIG start option in your start option list, to specify which definitions are to be activated at startup. In Figure 34 on page 135, for example, to use the configuration list defined in ATCCON01, you can specify CONFIG=01 on the VTAM START command.

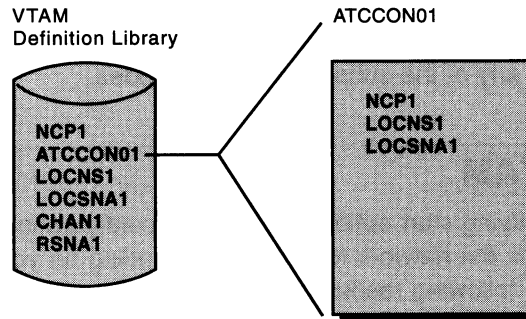


Figure 34. Configuration List

ATCCON01 contains the member names NCP1, LOCNS1, and LOCSNA1, which will be activated when VTAM starts. CHAN1 and RSNA1 are not included in ATCCON01, and therefore must be activated manually by the VTAM operator using the VARY command, if they are to be used. If no configuration lists exist, the VTAM operator must use the VARY command to activate all resources.

Following is sample coding for configuration list ATCCON01. Note the continuation characters in column 72.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
NCP1,                                                                    X
LOCNS1,                                                                    X
LOCSNA1
```

Following are examples of resource definitions that can be included in a configuration list:

- Major nodes
- Minor nodes defined in previously listed major nodes
- Paths
- Tables
- Dynamic reconfiguration files.

You can also use a configuration list to activate an NCP using a PU name which is different than the load module name (which is the same as the member name of the NCP in VTAMLST). For example, if you want to load a communication controller with the load module X3745A and you want the PU name to be NCPA1, you would code the following in your configuration list:

```
NCPA1/X3745A
```

This would have the same result as using the following command:

```
VARY NET ACT,ID=NCPA1,LOADMOD=X3745A
```

When specifying PU name/load module, if the BUILD statement in the NCP major node definition specifies a PUNAME, that same PU name must be used in the configuration list or the activation will fail. Likewise, if the BUILD statement specifies a NEWNAME, that same name must be used as the load module name in the configuration list or activation will fail.

Each of the names specified in the configuration list must be 1–8 characters long (1–7 characters for the load module name when specifying pu_name/load_module). The first character of each name must be alphabetical (A–Z) or one of the national characters (@, #, or \$). The remaining characters of each name must be alphabet-

ical (A–Z), numerical (0–9), or one of the national characters (@, #, or \$). If a name is found that is not valid, VTAM stops processing the configuration list and is unable to initialize any of the subsequent major nodes.

Identifying Resources to VTAM

In addition to specifying start options and coding configuration lists, you'll need to identify resources in the network to VTAM. Depending on your network, define a combination of the following resources:

- Application programs
- NCPs
- Peripheral nodes
- LUs
- Other VTAMs.

Coding Concepts

This section describes basic coding concepts needed to define resources. For further information, see the *VTAM Resource Definition Reference*.

To define your resources:

1. Code VTAM and NCP definition statements to define your major nodes.

VM Name your files *majornodename* VTAMLST. The files must have the characteristics of RECFM=F, LRECL=80.

The following restrictions apply to coding your definition statements:

- Names must be unique within a network.
- Names cannot begin with "IST" or any of the names "VTAM" "VTAMSEG," or "TRACE" (VSE only). Also, the name "VTAMTERM" cannot be used.
- Code each minor node and major node name in column 1.
- Code each definition statement in column 10.
- Code each operand in column 16 or one space after the definition statement.
- If your operands are longer than one line, code a continuation character in column 72 and continue operands in column 16 of the next line.

Following is a sample definition file KIMAP VTAMLST (the scale is included to show coding restrictions, such as the continuation character in column 72):

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
KIMAP  VBUILD TYPE=APPL
AP1    APPL  AUTH=(PASS,ACQ),                                X
        ACBNAME=ECHO
AP2    APPL  ACBNAME=ECHO
```

2. Code the NCP-only definition statements and operands as described in the NCP resource definition books, if you are defining NCP.
3. Generate the NCP as described in the *NCP, SSP, and EP Generation and Loading Guide*. If you are updating VTAM-only definition statements or operands, update the NCP definitions in VTAMLST; no NCP generation is required. If you are updating NCP-only definition statements or operands, update the

NCP generation definition and generate NCP again, or use dynamic reconfiguration facilities.

Note: Some dynamic VTAM facilities require no regeneration of the NCP.

4. Store your definition statements:

- **VM** The files are stored on a minidisk that is linked to the VTAM virtual machine. Whenever you add or change VTAM definitions you must use the GCS ACCESS command to reaccess the disk that contains the corresponding VTAMLST files before you activate the resources or restart VTAM.
- **MVS,VSE** Store your definitions in the proper VTAM library. You can concatenate VTAMLST files, but either the block sizes must be the same for all files or the file with the largest block size must be the first file in the concatenated list.

Sift-Down Effect

The “sift-down” effect enables you to code an operand on a higher-level node so that you do not need to recode it on each lower-level node for which you want the same value. As a result, the sift-down effect greatly simplifies the coding process.

The *VTAM Resource Definition Reference* identifies and describes the definition statements and operands to which sifting applies. For information about definition statement sequencing and the sift-down level for each NCP operand, refer to the NCP resource definition books.

Initial Status

Using VTAM definition statements and initial configuration lists, you can specify the initial status of each minor node in the domain as active or inactive. If a node is not activated when VTAM is started, the VTAM operator must activate it before a session can be initiated.

MVS,VSE If you are using a warm start of VTAM, the status recorded in a configuration restart file overrides initial status. VTAM either activates a node or leaves a node inactive according to the status of the node when VTAM was last running.

Certain internally maintained resources are automatically activated by VTAM when the message “VTAM INITIALIZATION COMPLETE” is issued. These resources can be displayed, but cannot be activated or deactivated by an operator. The following resources are automatically activated:

- VTAMSEG application program major node:
 - VTAM (or name from the CDRM definition statement for this VTAM)
 - ISTATA00
 - ISTNOP
 - ISTDCLU
 - ISTAPNCP
- ISTPUS PU (or name from HOSTPU start option) type 5 node
 - ISTGROUP
- ISTDILU predefined independent LU major node
- ISTDSWMN dynamic switched major node
- ISTADJCP adjacent CP major node
- ISTCDRDY dynamic cross-domain resource major node.

Note: The ISTCDRDY major node can be deactivated and activated by an operator. For further information, see “Dynamic Definition of Independent LUs” on page 221.

Defining Application Programs

Each application program must be defined within an application program major node. Each application program represents a minor node.

You define an application program major node with one VBUILD definition statement, and you define application program minor nodes with one APPL definition statement for each application program in the major node. Following is a sample of definition statements for an application program major node. Each APPL definition statement represents a single logical unit to VTAM.

```
APPLNODE VBUILD TYPE=APPL
A50PAYRL APPL  AUTH=ACQ,          PERMIT APPLICATION TO ACQUIRE LUS
                EAS=1000,         CONCURRENT APPLICATION SESSIONS
                MODETAB=COMTAB,    LOGON MODE TABLE=APPLICATION IS SLU
                APPC=YES,          ENABLE LU 6.2 SUPPORT
                PARSESS=YES,       ENABLE PARALLEL SESSION FOR LU 6.2 SUPPORT
                ACBNAME=PAYROLL,   NETWORK LU NAME=A50PAYRL
                                   VTAM APPLICATION APPLID=PAYROLL

:
A50ACCTS APPL  ATNLOSS=ALL,       ENABLE ATTN EXIT FOR ALL LU 6.2 SESSIONS
                APPC=YES,          ENABLE LU 6.2 SUPPORT
                PARSESS=YES,       ENABLE PARALLEL SESSION FOR LU 6.2 SUPPORT
                SECACPT=ALREADYV,  ACCEPT ALREADY VERIFIED LU 6.2 CONVERSATIONS
                ACBNAME=ACCOUNTS   NETWORK LU NAME=A50ACCTS
                                   VTAM APPLICATION APPLID=ACCOUNTS

:
```

For further information on application programs, see “Application Programs” on page 161.

Attaching NCP to VTAM

An NCP major node, which is a type 4 subarea node, is defined by filing NCP definition statements in the VTAM definition library. The name under which the major node is filed must match the NCP load module name on the NEWNAME operand of the BUILD definition statement.

Additional information required by VTAM to communicate with the NCP and its attached devices is provided to VTAM by coding VTAM-only definition statements and VTAM-only operands on NCP definition statements. The VTAM-only definition statements and operands must appear in the NCP definition statements that define this NCP to VTAM, even though they provide no information to the NCP. If VTAM-only operands are changed or replaced, no new NCP generation is required.

For further information on attaching NCP to VTAM, see Table 6 on page 139.

Table 6. Attaching NCP to VTAM

For information on:	See page:
NCP definition statements for VTAM	202
NCP definition statements for NCP	207
Channel-attached NCP	207
Link-attached NCP	212

Defining Peripheral Nodes and Logical Units

Peripheral nodes use local addresses for routing and require boundary function assistance from VTAM or the NCP. Logical units are the ports through which end users access the network. The type 1 and 2 peripheral node architecture supports dependent logical units only. The type 2.1 peripheral node architecture supports independent and dependent logical units.

Defining Type 2.1 Peripheral Nodes

For switched connections, PU definitions for type 2.1 peripheral nodes can be created dynamically on the VTAM host accepting the incoming call. For details, see “Defining Switched Resources Dynamically” on page 143.

If you code the PU definition statements for type 2.1 peripheral nodes, you might need to consider the following.

Type 2.1 Characteristics: Type 2.1 peripheral nodes can communicate with each other as well as with application programs in VTAM. Communication between type 2.1 peripheral nodes can occur through the main communication path among the subarea nodes in the network. Type 2.1 peripheral nodes support link-level role negotiation, eliminate hierarchical control (no ACTPU or ACTLU unless PU requests), require fewer flows to establish sessions, and use dynamically assigned session identifiers rather than pre-assigned addresses, all of which require fewer system definitions.

VTAM performs the directory services function for the subarea network and any APPN end nodes it serves. It locates the other session partner through a type 2.1 logical unit session request. VTAM is also involved in the session setup process when the type 2.1 session request traverses the subarea network, or when it is acting as a network node server for an APPN end node. VTAM or VTAM and NCP together appear and operate as a peer node to any type 2.1 peripheral node that attaches to the subarea network.

CPNAME Operand: For a type 2.1 peripheral node on a switched or LAN connection, VTAM requires the coding of either the CPNAME operand or both the IDBLK and IDNUM operands on the PU definition statement in a switched major node. You can code all three. During activation of the node, VTAM uses the NETID (if specified) and the control point name (CPNAME) to find the PU and LU definition statements associated with the node. If CPNAME is not in the XID or if CPNAME is not coded in VTAM, VTAM uses IDBLK and IDNUM instead. (To have VTAM first use IDBLK and IDNUM, use the SWNORDER=STATNID start option.) VTAM also uses the control point name to locate switched node definitions, even if the boundary function does not provide type 2.1 peripheral node support.

XID Services: For a type 2.1 peripheral node attached to the communication adapter using an SDLC link, the MODE operand on the LINE definition statement can be used to indicate whether VTAM performs the functions of the primary (PRI) or the secondary (SEC) link station. VTAM uses this PRI or SEC indication to determine its link-level role during the exchange identification (XID) processing for the connection. Because VTAM does not negotiate the primary or secondary role and uses the MODE specification for the XID process, you need to know the link-level role capability of the other type 2.1 connection. For example, if you specify VTAM as primary (MODE=PRI) and the other peripheral node is also primary non-negotiable, the connection fails.

If the SDLC line used is a switched link, the XMITDLY operand of the LINE definition statement can be used to specify the amount of time that VTAM should delay before performing XID processing after answering the incoming call. This delay allows the calling peripheral node to initiate XID transmission first and, therefore, to avoid VTAM initiating this process before the type 2.1 peripheral node is ready.

The XID=YES operand coded on the PU definition statement implies that the contact procedure used is for a type 2.1 peripheral node. VTAM determines whether the peripheral node is a type 2.1 or a type 2 by checking the XID value received from the node either during activation of the node or at CONTACT time for switched nodes.

ICA and XCA Local Area Networks: Although VTAM can support type 2.1 peripheral nodes that are channel-attached through the external communication adapter (XCA) or the integrated communication adapter (ICA), review the connectivity capabilities of each particular device to ensure it can be channel-attached as a type 2.1 peripheral node.

Defining a Type 2.1 Channel: If you are connecting a channel-attached type 2.1 PU to your host, code a VBUILD definition statement with TYPE=LOCAL specified.

```
LSNA3    VBUILD TYPE=LOCAL
LSNA3PU  PU    CUADDR=051,
              MAXBFRU=15,
              PUTYPE=2,
              SSCPFM=USSSCS,
              VPACING=0,
              XID=YES
```

Independent LUs

Independent logical units are represented to VTAM as cross-domain resources (CDRSCs). Independent LUs can be coded with LOCADDR=0 in major nodes, coded as CDRSCs, or dynamically defined, but in all cases, VTAM represents them as CDRSCs. The independent logical unit can access the network from multiple adjacent link stations, which can be in any domain. Even if you predefine independent LUs in device-type major nodes (NCP, local SNA, and switched major nodes), VTAM converts these definitions to CDRSCs when the major node is activated. Because independent LUs are represented as CDRSCs, when you use a DISPLAY command on an independent LU, the VTAM message responds with TYPE=CDRSC. However, message IST1131I shows a device type of INDEPENDENT LU / CDRSC.

For further information on independent LUs, see Table 7 on page 141.

Table 7. Independent LUs

For information on:	See page:
Characteristics of independent LUs	218
Defining independent LUs	220
Multiple connections between a type 2.1 node and a subarea node	224
Dynamic selection of session connections	225
Connecting independent LUs through NCPs	226
Transmission priority for independent LUs	227
Restrictions on using independent LUs	227

Defining Dependent Logical Units

Dependent logical units:

- Generally have unique, fixed, pre-assigned local addresses within the physical unit that are specified on the NCP or VTAM definition statements using the LOCADDR operand of the LU definition statement. With dynamic definition and with switched connections, the address is defined during the PU and LU activation process. For more information, see “Defining Dependent LUs Dynamically.”
- Are subordinate to a PU. If defined manually, they must be defined under a PU.
- Require an SSCP-LU session and an SSCP-PU session to be active to VTAM; these sessions must be established before any LU-LU session can be activated.
- Normally act as secondary logical units only.
- Can have only a single session. Even a logical unit that supports LU 6.2 protocols can have only one session if it is defined as a dependent logical unit.

Ownership of dependent logical units is determined when the SSCP establishes an SSCP-LU session. A dependent logical unit must be activated, controlled, and owned by a VTAM SSCP.

Defining Dependent LUs Dynamically

You can enable VTAM to define dependent logical units dynamically. One way to do this is to use the IBM-supplied selection of definitions for dependent LUs (SDDLUI) exit routine. With this exit, VTAM can define logical units dynamically when it receives information from the PU that the device is powered on, rather than when the major node is activated. The dependent logical units must be attached to the host through physical units that support Reply/PSID NMVTs, request units that are used for dynamic definition of dependent logical units. For example, you can use a 3174 cluster controller because it supports these request units.

VTAM uses model LU definition statements to build LU definitions. Define model LU definition statements based on information the physical unit sends to VTAM about the logical unit's device. The default information that the physical unit sends is the 7-character machine type and model number. When the physical unit is activated or when a device powers on, the physical unit sends a Reply/PSID NMVT request unit to VTAM, which can trigger VTAM to build or change LU definitions.

Attaching Peripheral Nodes to NCP

This NMVT contains the local address of each LU, a power indicator (on or off), the machine type and model number of the device, and optionally other device-dependent information needed to define the logical units. VTAM uses this information to choose an appropriate model LU definition statement to build an LU definition.

For details on defining dependent LUs dynamically, see “Defining Dependent LUs Dynamically” on page 216.

Attaching Peripheral Nodes to VTAM

Peripheral nodes can be attached to VTAM by a channel or an external communication adapter (XCA), the 3172 interconnect controller. The following channel connections can be used to attach a peripheral node to VTAM:

- Local non-SNA
- Local SNA
- Loop-adapter-attached.

Peripheral nodes can attach to the following types of local area networks (LANs) and connect to VTAM through an external communication adapter:

- Token ring
- Fiber Distributed Data Interface (FDDI)
- Carrier sense multiple access with collision detection (CSMA/CD) 802.3.

For further information on attaching peripheral nodes to VTAM, see Table 8.

Table 8. Attaching Peripheral Nodes to VTAM

For information on:	See page:
Local non-SNA connection	228
Local SNA connection	229
Loop-adapter-attached connection	231
External communication adapter (XCA) connections	231

Attaching Peripheral Nodes to NCP

The following connections can be used to attach a peripheral node to NCP:

- Nonswitched SDLC lines
- Switched SDLC lines
- NCP Token-Ring interconnection (NTRI)
- NCP Packet Switching Interface (NPSI)
- Binary synchronous (BSC) lines.

Note: Start-stop support is defined to VTAM as SNA physical and logical units. The start-stop protocol must be converted or enveloped in an SNA data stream using either a 3708 protocol converter or the NTO program product.

For further information on attaching peripheral nodes to NCP, see Table 9 on page 143.

Table 9. Attaching Peripheral Nodes to NCP

For information on:	See:
Defining type 2.1 peripheral nodes	"Defining Type 2.1 Peripheral Nodes" on page 249
Non-native network type 2.1 connections	"Non-Native Network Type 2.1 Connections" on page 250
Nonswitched SDLC connection	"Nonswitched SDLC Connection" on page 250
Switched SDLC connection	"Switched SDLC Connection" on page 251
NCP/Token-Ring interconnection (NTRI) connection	"NCP/Token-Ring Interconnection (NTRI)" on page 256
NCP Packet Switching Interface (NPSI) X.25 connection	<i>X.25 Network Control Program Packet Switching Interface Planning and Installation</i> "NCP Packet Switching Interface (NPSI) Connections" on page 259
BSC connection	"BSC Connection" on page 259

X.21 Connection (VSE): For NPSI connections, X.21 short hold mode/multiple port sharing (SHM/MPS) is defined on the PATH definition statement using the SHOLD, SHM, and SHMTIM operands. See the *VTAM Resource Definition Reference* for an explanation of these operands.

X.21 SHM/MPS is supported by coding the SHM and X21SW operands on the GROUP definition statement. Only lines that represent SVCs controlled by NPSI are capable of X.21 SHM/MPS.

You can specify that some lines be used for direct call only. Code AUTODL=NO on the GROUP or PU definition statements in the NCP major node. Then, when VTAM searches a line group for an available line, it cannot select a direct call only line for an auto-call operation.

VTAM does not need to know the telephone numbers. Instead, the PATH definition statement in the switched major node specifies the switched line to be used, and the X.21 network establishes the connection automatically when requested.

If an X.21 direct call line is available, its PATH definition statement should immediately follow the switched physical unit's PU definition statement. This ensures that VTAM attempts to call the physical unit on the direct call line before trying a dial line.

Defining Switched Resources Dynamically

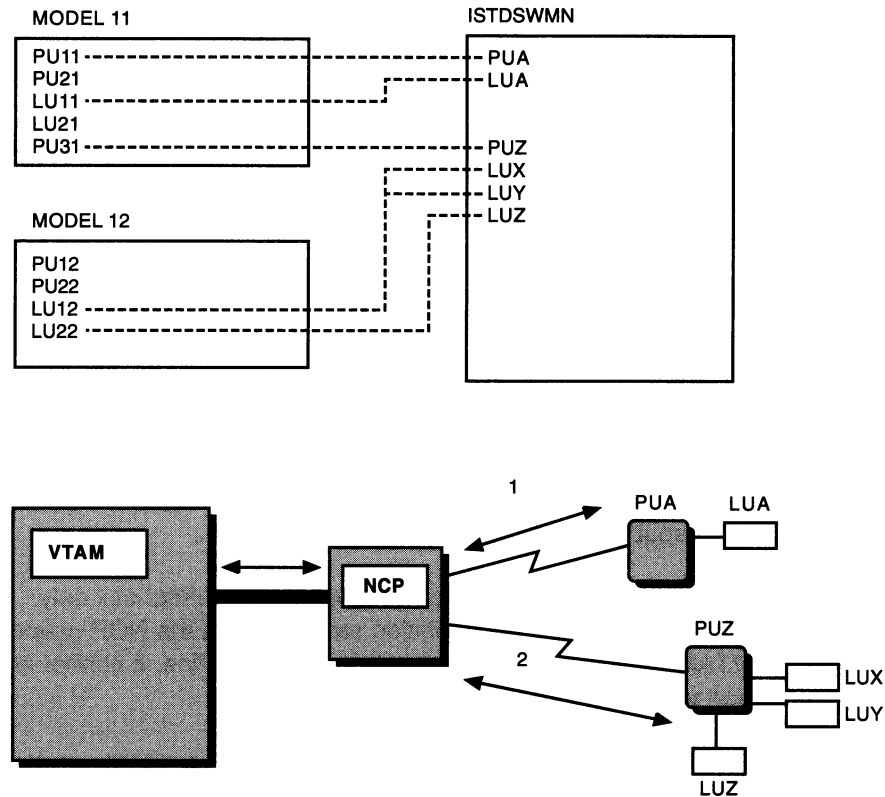
You can dynamically define switched peripheral nodes (including LAN-attached nodes) using:

- Dynamic PU definition (DYNPU operand)
- The dynamic switched definition facility.

Dynamic PU definition enables PU definitions for type 2.1 peripheral nodes to be created dynamically on the VTAM host accepting an incoming call. To use dynamic PU definition, simply code DYNPU=YES on the GROUP definition statement in an NCP, external communications adapter, channel attachment (ICA

SDLC), LAN, or packet major node. Then, if a calling PU is not already defined to VTAM, a PU definition is created for it.

The dynamic switched definition facility requires model definition statements and an exit routine. Model definitions enable you to create a major node from which definitions are taken to build dynamic switched devices. When a device dials in, a configuration services XID exit routine is invoked with the IDBLK, IDNUM, and other parameters from the device. The exit returns to VTAM the model statements and the names of the new PUs and LUs to use. VTAM then builds the new devices in the dynamic switched major node (ISTDSWMN) using the model definition statements. Figure 35 illustrates this process.



1. Device PUA dials in to VTAM. PU and LU definitions from MODEL 11 are used to create resources in the dynamic switched major node (ISTDSWMN).
2. Device PUZ dials in to VTAM. PU and LU definitions in MODEL 11 and MODEL 12 are used to create resources in ISTDSWMN.

Figure 35. Creating Resources in Dynamic Switched Major Node

For details on the use of the dynamic switched definition facility, see “Dynamically Defining Switched Resources” on page 260.

Note: If a configuration services XID exit routine is active, it is used to define a PU regardless of the DYNPU value coded on the GROUP definition statement.

Attaching Other Domains to VTAM

If a VTAM is connected to another VTAM, a physical link between them must be connected and functional at both ends. Each VTAM controls and must activate one end of the connection.

For information about connecting two or more domains, see Table 10.

Table 10. Attaching other domains to VTAM

For information on:	See page:
VTAM-to-VTAM channel connections	262
VTAM-to-VTAM NCP connections	267
VTAM-to-VTAM external communication adapter (XCA) connections	274
VTAM-to-VTAM integrated communication adapter (ICA) connections.	278

Sharing NCP Peripheral Nodes (CMC Configurations)

When you plan to share NCP resources:

- Provide a data host to back up a communication management configuration (CMC) host.
- Code the PCCU definition statements in the NCP definitions filed at the CMC hosts.
- Code the HOST definition statements in the NCP definitions filed at the CMC hosts.

Figure 36 on page 146 shows a possible CMC. In this figure, NCP1 is channel-attached to HOST1, but it is in HOST2's domain. NCP1 is a channel-attached cross-domain NCP with respect to HOST1. The same relationship applies between NCP3 and HOST3. All three NCPs are controlled by HOST2, the CMC host. HOST1 and HOST3 can contact their channel-attached NCPs and communicate with resources attached to them without ever activating those NCPs, which is normally referred to as a data host connection. CMC HOST2 is responsible for activating all three NCPs. You must define the channel connection between the data hosts and NCPs in a channel attachment major node. However, the VTAMs in the data hosts do not require the NCP load module, the NCP resource resolution table (RRT), or any NCP major node definitions in VTAMLST. All network problems are handled by HOST2. HOST1 and HOST3 (designated as data hosts) do not own any of the network resources. All LU-LU sessions to remote network resources are cross-domain sessions.

Note: The NCP must be loaded and active if VTAM is to contact the NCP but not establish an SSCP-PU session with it.

If the NCP is deactivated with the VARY INACT command at the CMC and the channel-attached major node is active, sessions using the channel path to the data host continue uninterrupted.

The channel connection between HOST2 and NCP1 is defined to HOST2 in a channel-attachment major node. The normal operating activation procedure for a CMC configuration consists of the following steps:

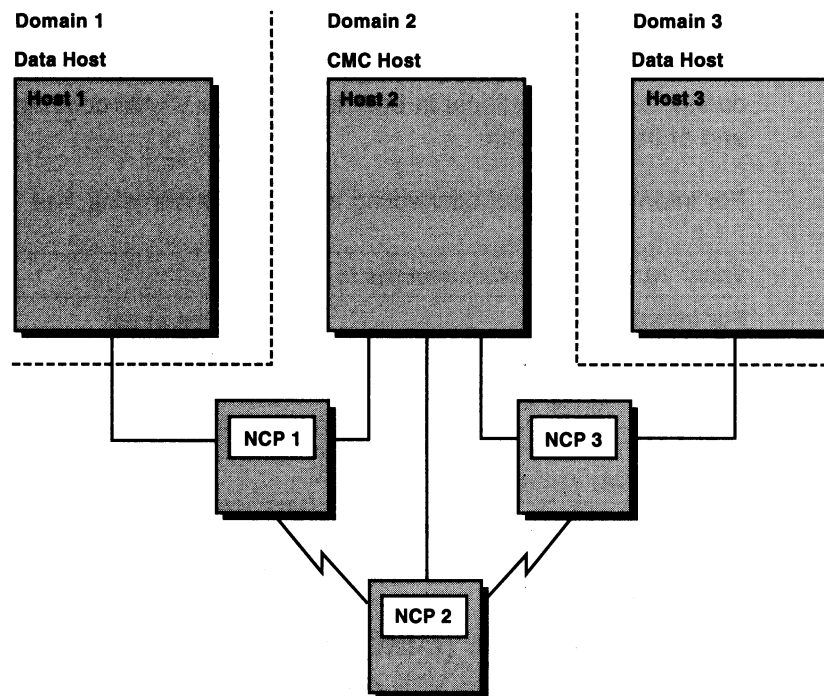


Figure 36. Example of Communication Management Configuration

1. HOST2 activates NCP1.
2. HOST1 activates its channel-attachment major node to contact NCP1.

If CMC HOST2 fails and must be backed up by data HOST1, HOST1 activates NCP1. In this case, HOST1 needs a copy of the NCP source and RRT. Activating the NCP establishes an SSCP-PU session between HOST1 and NCP1; HOST1 becomes the new CMC host. HOST3 can provide backup for HOST2 in much the same way. However, for either data host to back up the CMC, VTAM must be capable of accessing the NCP major node definitions in VTAMLST and a copy of the NCP RRT. If either data host also needs to load or dump the NCPs in backup situations, VTAM needs access to the NCP load modules, SSP, and dump data sets.

For further information on CMC configurations, see Table 11.

Table 11. CMC configurations

For information on:	See page:
Channel-attached NCPs in a CMC configuration	284
Link-attached NCPs in a CMC configuration	286
Controlling resource ownership in a CMC configuration	286
Problem determination in a CMC configuration	287

Dynamic Configuration of Channel-Attached Devices (MVS)

Dynamic configuration of channel-attached devices (dynamic I/O) enables you to modify your I/O configuration without disrupting your system. You can add, delete, or modify I/O components, such as devices and paths, without having to re-IML or re-IPL.

Use the following to dynamically configure your channel-attached devices and dynamically create VTAMLST members for new devices:

- MVS V4R2 or later
- Hardware configuration definition (HCD)
- NetView Release 3 and subsequent releases, or a product with a similar interface and services.

Note: NetView and TSO/E are necessary for dynamically creating VTAMLST members using the VTAM-supplied sample command lists. To modify the I/O configuration only, you need only MVS and HCD.

The following list enumerates how a channel-attached device is added in Figure 37 on page 148, using the dynamic I/O configuration support.

1. Attach a device physically.
2. Add the device to your configuration using hardware configuration definition (HCD) and activate the new configuration using the MVS ACTIVATE command.
3. MVS issues message IOS502I, which causes NetView to start the IOS502I command list.
4. The IOS502I command list calls the ISTDEFIN command list for each device added to the system.
5. The ISTDEFIN command list obtains a description of the new device using the ISTDINFO function module, which obtains the new device information and returns it to ISTDEFIN.
6. The ISTDEFIN command list builds a VTAMLST member for the device. Any information the device cannot provide is supplied with IBM- or user-supplied defaults.
7. The ISTDEFIN command list issues a VARY ONLINE command for the device if it is not already online.
8. The ISTDEFIN command list issues a VARY ACT command to activate the device.

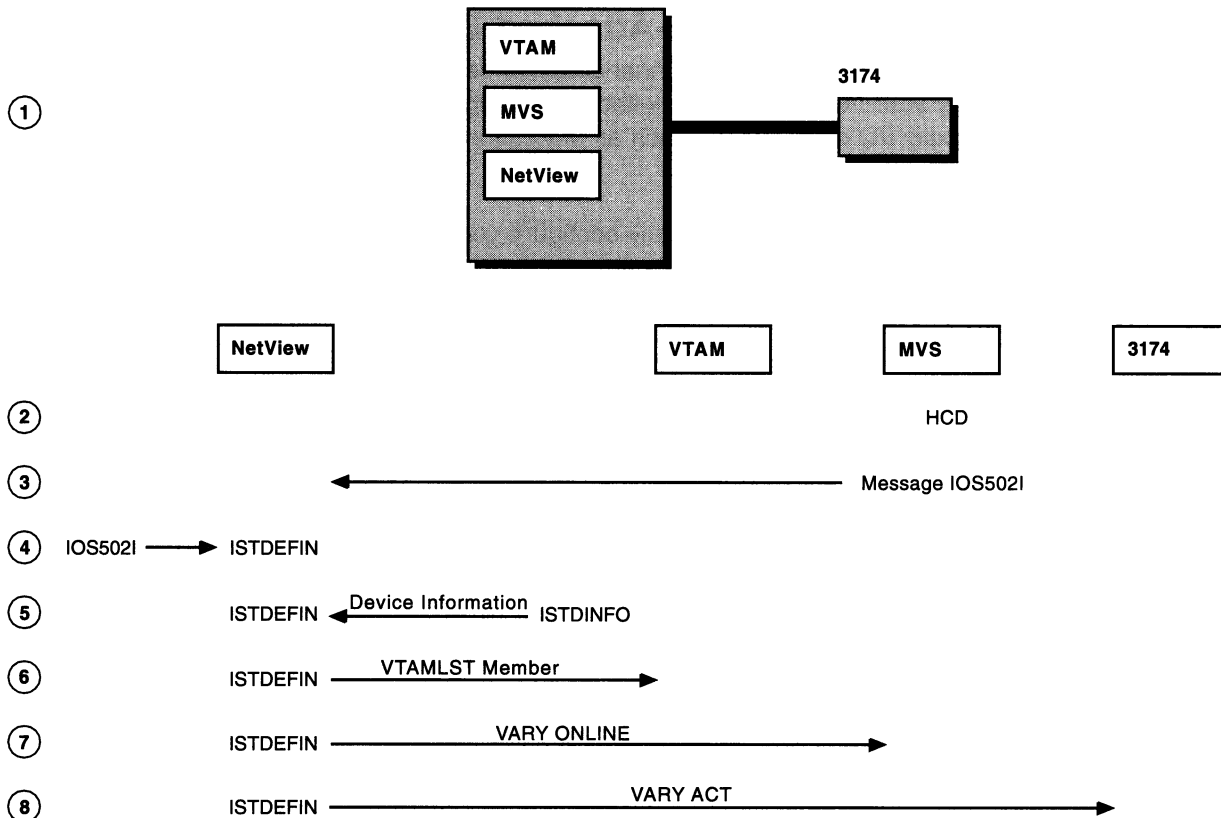


Figure 37. Dynamic Configuration of Channel-Attached Device

The following tasks are necessary to dynamically configure channel-attached devices:

Installation and preparation

For information on the installation steps in NetView, see “Installation and Preparation” on page 287. Also add data sets; see “Data Sets for Dynamic I/O Configuration” on page 27.

Defining your configuration

For a description of how to define your configuration, see “Defining Your Configuration” on page 288.

Building resource definitions

For a description of how ISTDEFIN defines and activates the devices, see “Building Resource Definitions” on page 288.

Using the default naming convention

For a description of the IBM-supplied naming convention, see “Using the Default Naming Convention” on page 289.

Customizing the command lists

The IOS502I and ISTDEFIN command lists can run without any modification. However, you can modify the command lists to match the needs of your environment. For details, see “Customizing the Command Lists” on page 290.

Dynamic Reconfiguration and Change of Operands

Dynamic reconfiguration is the process of adding, deleting, or moving resources within your network configuration without deactivating the affected major node. For example, resources on an NCP can be changed without deactivating the NCP. Dynamic reconfiguration does not replace NCP generations, but supplements them to allow quick and efficient control of changes in the domain configuration. Dynamic reconfiguration is also possible for local peripheral nodes.

Dynamic change of operands enables you to modify certain operand values on PU and LU definition statements without deactivating the major node.

Note: LUs and PUs can be defined dynamically, in which case no system definition is required.

Following are the resources for which dynamic reconfiguration and dynamic change of operands are supported:

- PUs and LUs on nonswitched SDLC lines or ESCON * channels in an NCP major node
- LUs in a local SNA major node
- Type 1 PUs under NCP frame relay lines, and their associated FRSESET definition statements
- TRLEs in a TRL major node
- **VM,VSE** LUs under an SDLC nonswitched line in a channel-attachment major node
- **VM,VSE** LUs in an X.25 permanent virtual circuit in a packet major node.

Dynamic change of operands is also supported for PUs and LUs in switched major nodes.

Dynamic reconfiguration, unless noted below, requires NCP V4R3 or V5R3, or later, and allows you to:

- Add and delete type 1, type 2, and type 2.1 physical units and their associated logical units on nonswitched SDLC lines
- Add and delete logical units under physical units on nonswitched SDLC lines
- Add and delete type 2.1 physical units on an NCP ESCON channel
- Add and delete independent logical units on an NCP ESCON channel

Note: Independent LUs can be dynamically defined, in which case no system definition of the independent LU is required. Independent LUs can also be defined using CDRSC definition statements.

- Add and delete NCP frame relay type 1 PUs and their associated FRSESET definition statements
- Move a physical unit and its associated logical units from its current line to another line within the same NCP
- Move a logical unit to another physical unit on non-switched SDLC lines
- Change the ADDR for a physical unit on non-switched SDLC lines
- Change the LOCADDR for a logical unit on non-switched SDLC lines

Identifying Routes and Establishing Sessions

- Add TRLE entries to a TRL major node or replace existing TRLE entries that are not currently in use (that is, a TRLE entry that is currently in use is not deleted; new entries are still added).

Note: NCP V4R3 or V5R3, or later, is not required for dynamic reconfiguration of TRLE entries.

There are several ways to do dynamic reconfiguration:

- Add, delete, or move resources in the VTAMLST definition file and issue a VARY ACT,UPDATE=ALL command. Changes made in this way are permanent changes, and do not require the major node to be inactivated.
- Code a DR file to add, delete, and move resources and activate it with the VARY DRDS command. Changes made in this way are temporary changes. If you want the changes to be permanent, a new NCP should be generated and reloaded to reflect the changes as soon as time permits.
- Use the MODIFY DR command to move or delete resources. Changes made in this way are temporary changes. If you want the changes to be permanent, a new NCP should be generated and reloaded to reflect the changes as soon as time permits.
- Add resources to the NCP's VTAMLST definition file, and then reactivate the NCP (implicit dynamic reconfiguration). This procedure is disruptive in that inactivation of an NCP causes all existing sessions to be terminated. As soon as possible, you should regenerate and reload the NCP to reflect the changes.

For further information on dynamic reconfiguration and change of operands, see Table 12.

Table 12. Dynamic Reconfiguration and Dynamic Change

For information on:	See page:
Dynamic reconfiguration and dynamic change requirements	291
Using the VARY ACT, UPDATE technique	292
Using the VARY DRDS technique	295
Using the MODIFY DR technique	297
Dynamic reconfiguration of independent LUs	297
Dynamic reconfiguration of frame relay resources	298
Implicit dynamic reconfiguration	300

For some examples of dynamic reconfiguration, see “Dynamic Reconfiguration Examples” on page 301.

Identifying Routes and Establishing Sessions

In addition to using start options and configuration lists and defining resources, routes must be identified to enable resources to establish sessions with each other. This section provides an overview of the following topics:

- Network Routing for Subarea Nodes
- Establishing and Controlling SNA Sessions.

Network Routing for Subarea Nodes

You need to design routes between:

- VTAM and any NCPs it activates
- NCPs communicating over a subarea connection (which includes NCPs within a composite network node)
- VTAMs communicating over a subarea connection.

It is important that you consider the topics in Table 13 when designing routes in your network. Following Table 13 is information on the fundamentals of paths and network routing.

Table 13. Network Routing for Subarea Nodes

For information on:	See page:
How VTAM handles network and subarea addressing	306
Compatibility between non-extended and extended network addressing nodes	307
Compatibility between extended network and extended subarea addressing nodes	309
Virtual route pacing	310
Parallel sessions using parallel transmission groups	312
Dynamic path update	313

Communication between two network accessible units (NAUs) in different subareas requires a definition of at least one route connecting the subareas. This definition includes a physical and logical path between the two.

Physical Paths: The physical path between two subarea nodes is an explicit route.

explicit route (ER)

An explicit route is an ordered set of subarea nodes and transmission groups along a path between communicating subarea nodes, including:

- The endpoint subareas
- Any subareas between the endpoint subareas
- The transmission group used to connect each subarea pair along the route.

In connecting one endpoint subarea to another, an explicit route cannot include the same subarea more than once.

forward explicit route

Explicit routes originating in the host are forward explicit routes and are numbered 0 through 15 (0 through 7 with releases prior to NCP V4R3.1 or V5R2.1 or if ERLIMIT is set to 8).

reverse explicit route

Reverse explicit routes terminate in the host and must use the same set of subarea nodes and transmission groups as their corresponding

forward explicit route. They are also numbered 0 through 15 (0 through 7 with releases prior to NCP V4R3.1 or V5R2.1 or if ERLIMIT is set to 8), but they do not have to have the same explicit route number as the corresponding forward explicit route.

transmission group (TG)

A transmission group consists of a single SDLC link or data channel between two subarea nodes, or of a logical combination of up to eight parallel SDLC links between communication controller subarea nodes. Transmission groups are numbered 1 – 255.

Transmission groups appear as one link to the path control network, even if the transmission group contains parallel SDLC links. Data flows across a specific transmission group during a session, but not across a specific link in a transmission group. The specific link used within the transmission group is determined dynamically depending on which link is available at the time.

Transmission groups can also exist in parallel, creating multiple explicit routes between two subareas. For instance, parallel SDLC links between adjacent communication controller subarea nodes can be divided into multiple transmission groups. A maximum of 16 parallel transmission groups are allowed between adjacent subarea nodes.

Parallel links in one or more transmission groups can be active and carry data simultaneously. Each parallel link is controlled (activated and deactivated) independently of the others.

routing tables

An entire explicit route is not defined to each subarea node along a path; instead, each subarea node has its own routing table that contains only the information that path control needs to forward data to the next subarea along the route (the adjacent subarea). Given the explicit route number being used and the destination subarea, the table provides the adjacent subarea to which the data is forwarded and the transmission group over which the data is forwarded.

peripheral link

A peripheral link is the portion of the route between a subarea node and a peripheral node. A peripheral node uses local addresses for routing and requires boundary function assistance from an adjacent subarea node to communicate with a non-adjacent subarea node.

In the sample network configuration shown in Figure 38 on page 153, HOSTA and peripheral node T1 can communicate over two explicit routes, one in each direction. Both explicit routes have been numbered 0 and include HOSTA, NCPA1, NCPA2, and transmission groups 1 and 14. A data channel assigned to TG1 connects HOSTA with NCPA1. An SDLC link assigned to TG14 connects NCPA1 and NCPA2. Each subarea node contains a routing table to route data between subareas along ER0.

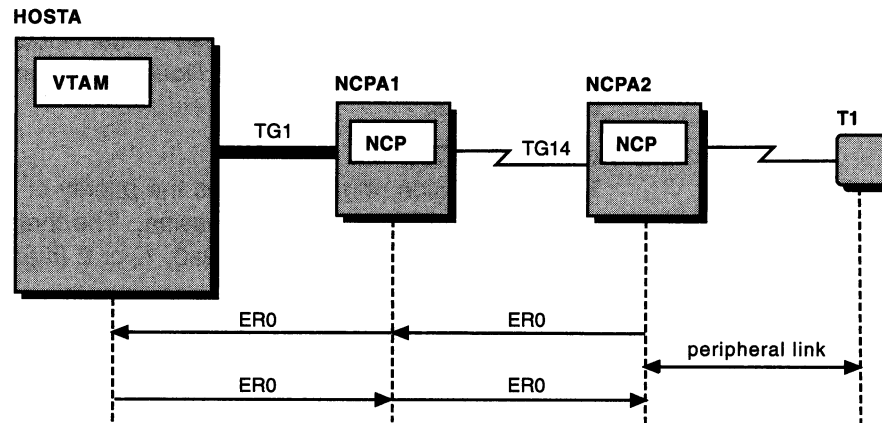


Figure 38. One Explicit Route in Each Direction

In the sample network configuration in Figure 39, parallel SDLC links connect NCPA1 and NCPA2. Two different explicit routes between HOSTA and NCPA2 are now available in either direction. All routes take TG1 between HOSTA and NCPA1, but data flows between NCPA1 and NCPA2 over TG14 for ER0 or TG15 for ER1. The routing tables have been updated to reflect the additional explicit routes.

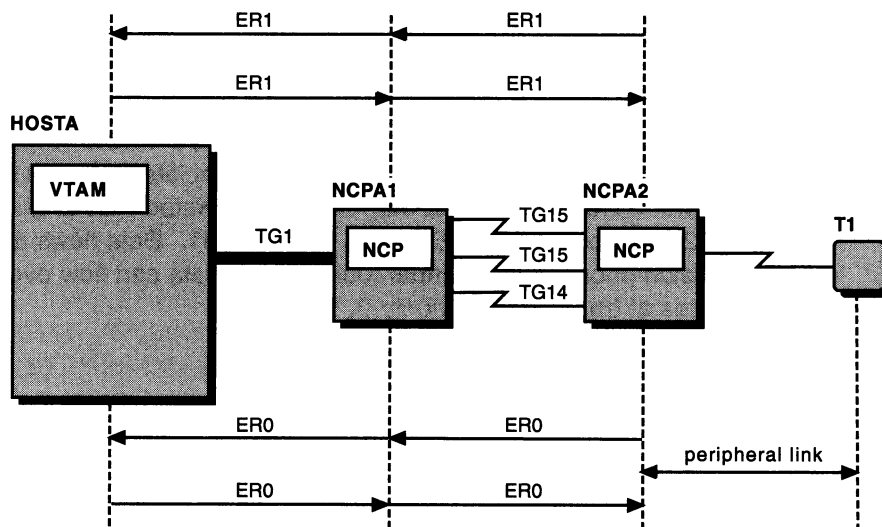


Figure 39. Two Explicit Routes in Each Direction

Logical Paths: The logical path between two subarea nodes is a virtual route.

virtual route (VR)

A virtual route is a bidirectional logical connection between two subarea nodes. At least one end of a virtual route must be in a subarea node that activates virtual routes. All hosts can activate virtual routes. NCP V4R2 and later releases can activate virtual routes.

Eight virtual routes numbered 0 to 7 can be defined between two subarea nodes.

One or more virtual routes must be defined for each forward-reverse explicit route pair. A virtual route places a transmission priority on data traffic using the underlying explicit routes.

You can modify the list of virtual routes and their associated transmission priorities using the virtual route selection function of the session

management exit routine, or using the virtual route selection exit routine. For information on using these, see “Virtual Route Selection Function (Function Code X'0B')” or “Virtual Route Selection Exit Routine” in *VTAM Customization*.

transmission priority (TP)

The transmission priority identifies the priority of message units flowing over an explicit route during a session. The three possible levels of transmission priority are: 0 (lowest), 1, or 2 (highest).

In general, high priority messages are routed before low priority messages. Within a specific transmission priority, messages are routed on a first-in-first-out (FIFO) basis.

The eight virtual routes and three levels of transmission priority provide for the possibility of 24 virtual route and transmission priority pairs between two subarea nodes. In other words, each subarea pair can have virtual routes numbered 0 through 7 between them, and each virtual route can have up to three different transmission priorities.

route extension

A route extension is a logical connection between a subarea node and a peripheral node. A peripheral node uses local addresses for routing and requires boundary function assistance from an adjacent subarea node to communicate with a non-adjacent subarea node.

As shown in Figure 40 on page 155, a virtual route must be defined for each explicit route pair. For example, between HOSTA and NCPA1, VR0 is associated with ER0 and reverse ER0, and VR1 is associated with ER1 and reverse ER1. Between HOSTA and NCPA2, VR0 is associated with ER0 and reverse ER0, and VR2 is associated with ER1 and reverse ER1. Data flows at a designated transmission priority over a virtual route. Here, data can flow over any of the virtual routes at transmission priority 0, 1, or 2.

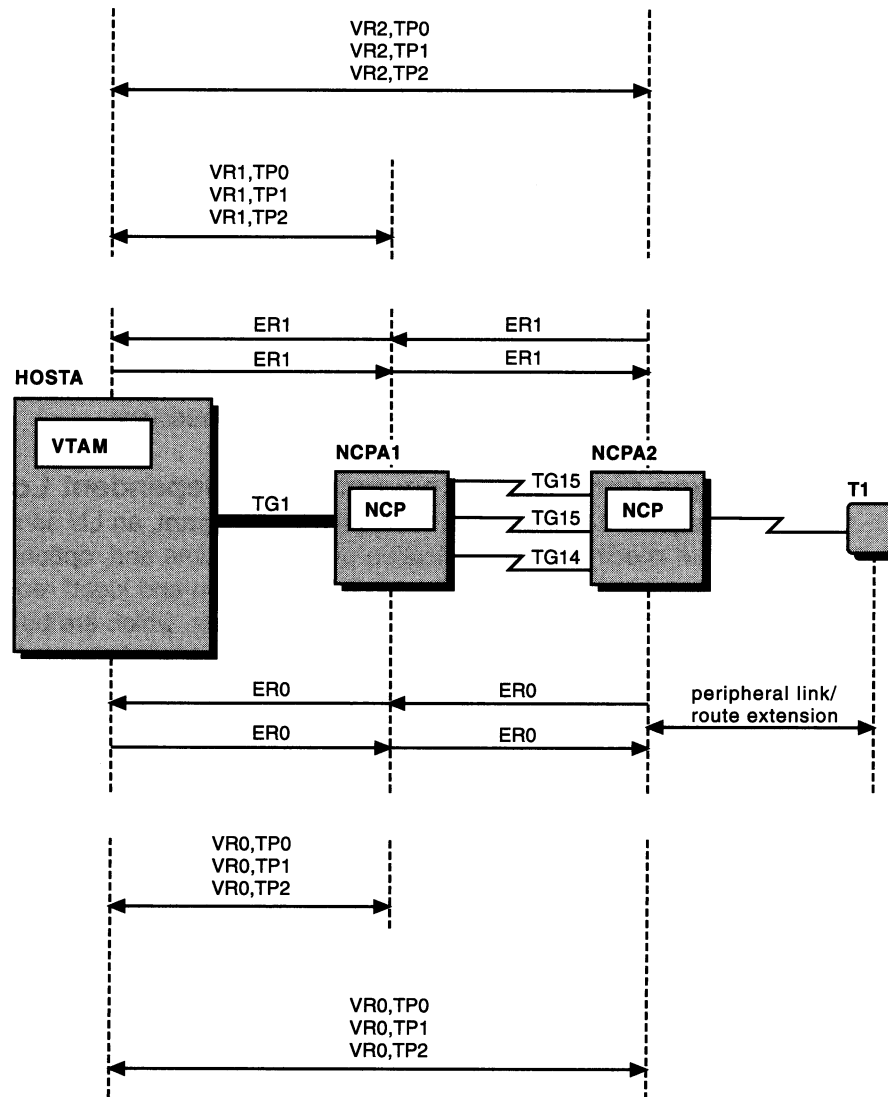


Figure 40. Virtual Route and Explicit Route Associations and Transmission Priority

Establishing and Controlling SNA Sessions

Logical units are defined to VTAM to provide access services to network resources. To tailor sessions, VTAM uses the unformatted system service (USS), interpret, logon mode, and class-of-service tables. VTAM also uses the logon-interpret routines and the virtual route selection and session management exit routines to authorize, establish, or terminate sessions. You can replace or modify these tables and routines. This, along with operands that can be specified in VTAM definitions, allows VTAM to adapt to your needs. The *VTAM Resource Definition Reference* describes how to replace or modify the tables used in session establishment and termination and how to specify the network definition operands that relate to sessions. *istprg0t* describes how VTAM initiates, controls, and terminates sessions. *VTAM Customization* describes the exit routines.

Sessions between logical units can be initiated by the following:

- Session-initiation requests from a logical unit
- Automatic logon

- The application program, using API requests (for example, OPNDST, SIMLOGON, or REQSESS)
- The VTAM operator, using VARY LOGON commands.

Normally, sessions are terminated by one of the following:

- Session-termination requests from a logical unit
- The application program, using API requests (for example, CLSDST or TERMSESS)
- The VTAM operator, using VARY resource inactivation commands.

VTAM Programming describes the use of VTAM application program interface (API) macroinstructions to establish and terminate sessions.

Logon and Logoff Requests from Dependent Logical Units

To request a session with an application program, an LU sends a logon request to VTAM specifying the application program's name and, optionally, a logon mode name and some additional information. Logon and logoff requests can be field-formatted SNA Initiate and Terminate requests, which are built by the logical unit that sent them, or they can be (unformatted) character-coded commands, which are translated by VTAM into a field-formatted SNA request. VTAM uses session-level USS tables and interpret tables to do the translation.

If a character-coded command follows the USS command syntax rules (see the *VTAM Resource Definition Reference*), then a session-level USS table can be used to convert the character-coded command into a field-formatted SNA request. These commands are referred to as USS commands. If a character-coded command violates these syntax rules, an interpret table must be used for the conversion.

You can customize VTAM to accept other character-coded commands by coding an interpret table. The following forms of requests make use of the interpret table to determine the name of the application program to which the request is being made:

- A field-formatted SNA request received directly from a logical unit
- An unformatted character-coded command, converted to a formatted SNA request.

For example, any of the above forms of a request can indicate a logon to APPL01, and the interpret table can change APPL01 into TESTAPL1.

You associate an interpret table to a logical unit using the LOGTAB operand on the LU definition statement. You associate a USS table to a logical unit using the USSTAB operand on the LU definition statement.

Note: For a complete description of operation- and session-level USS tables, see “Unformatted System Services (USS) Tables” in the *VTAM Resource Definition Reference*.

National Language Support

With national language support (NLS) you can tailor messages and commands for end users. Using the MVS Message Service (MMS) or the LANGTAB USS tables, you can define USS messages in several different languages.

Also, by default, the language passed to a VTAM application program comes from the logon mode entry specified by the USS LOGMODE parameter. With national language support, the language passed to the application program can be specified by the end user, independent of the LOGMODE parameter.

For further information on NLS, see “National Language Support for End-User USS Messages and Commands” on page 316 and “National Language Support for the Language Passed to Application Programs ” on page 317.

Model Name Table

The model name table contains model names that can be passed to VTAM application programs in their LOGON exits. The purpose is to help VTAM application programs to create dynamic definitions for their session-partner resources.

A model definition is normally used as the starting point for a dynamic resource definition. A typical application might maintain many different model definitions, one for each possible combination of resource characteristics. The model name passed by VTAM enables the application program to simply and accurately select the proper model definition to use.

Operands on an SLU's resource definition can associate that SLU with the proper model name data. The MDLTAB operand specifies the model name table to be used, and the MDLENT operand specifies the proper entry within the table.

For further information on the model name table, see “Model Name Table” on page 318.

Associated LU Table

An associated LU table contains associated LU names that can be passed to VTAM application programs in their LOGON exits. The purpose is to help VTAM application programs to create dynamic definitions for their session-partner resources.

The associated LU names provide supplementary information unavailable from model definitions. The LU names specify primary and alternate printers logically related to the SLU. (For information on model definitions, see “Model Name Table.”)

Operands on an SLU's resource definition can associate that SLU with the proper associated LU data. The ASLTAB operand specifies the associated LU table to be used, and the ASLENT operand specifies the proper entry within the table.

For further information on the associated LU table, see “Associated LU Table” on page 319.

Selecting Session Parameters for the Logon Mode Table

When a logical unit requests a session with an application program, it uses a symbolic logon mode name, either directly or by default, to suggest the session protocols. Session protocols are a set of rules that describe how the session is conducted. For example, one protocol might specify that the application program use chaining for SNA requests. Another might require that the logical unit not send end-bracket indicators to the application program. For a complete description of the session protocols available in VTAM, see “Defining and Naming a Set of Session Parameters (Logon Mode and Class of Service)” in *VTAM Programming*.

Session protocols are expressed as a string of characters called session parameters, usually specified in the logon mode table. A set of session parameters for a session is also called its logon mode. VTAM contains an IBM-supplied logon mode table, ISTINCLM. This table contains a set of generally accepted session parameters for a basic list of IBM device types, but it might not completely meet your needs. You can create supplementary logon mode tables and associate them with device type logical units or application programs using the MODETAB operands on the definition statements defining them.

For further information on logon mode tables and specifying session parameters, see “Selecting Session Parameters for the Logon Mode Table” on page 320.

Automatic Logons

You can enable your LUs to automatically logon to a particular application program. This is most useful when a designated terminal always accesses one application program. To enable automatic logons, you can either code the LOGAPPL operand on the LU definition statement, or you can use the VARY LOGON or VARY ACT,LOGON commands.

Automatic logon is useful for the following:

- Unattended terminals (terminals with no operator to issue a logon)
- Establishing a controlling application program to:
 - Manage allocation of terminals
 - Automatically re-establish sessions with terminals after failure and restart of a host or NCP.

For further information on automatic logons, see “Automatic Logons” on page 323.

Session Authorization and Accounting

The session management exit routine can be used to:

- Check or restrict the use of a logical unit (session authorization)
- Gather accounting information (session accounting).

Code a session management exit routine if you want to use these functions. There is no default session management exit routine. If you do not code a session management exit routine, it is assumed that all session requests are authorized and no accounting data is collected by VTAM.

For further information on session authorization and accounting, see “Session Authorization and Accounting” on page 329.

Pacing and Flow Control

The number of active application programs and the amount of data they send and receive varies. Because of this, the amount of data flowing through a network often varies. In many networks, you can anticipate and plan for the peak traffic periods. However, unexpected peak traffic demands can result from conditions such as the failure of a route and the resulting diversion of its traffic to other routes. A route can become overloaded with the extra traffic it is asked to carry. Unexpected demands by network users for data transmissions can impose unusual peak traffic conditions on parts of the network. Thus, the distribution of the traffic in various parts of the network and the total traffic in a network might vary.

Data congestion results whenever the rate at which data is going into a network exceeds the capacity of the network routes over which the data must flow. Response times might lengthen and throughput might decrease. Severe or prolonged congestion in one part of a network can affect the other parts, causing overall network efficiency to suffer.

Both VTAM and NCP monitor traffic and limit congestion in the network by means of a flow control technique called pacing. Two kinds of pacing are involved in the flow-control process: session pacing and virtual route pacing. For information on session pacing, see “Pacing and Flow Control” on page 330. For information about virtual route pacing, refer to “Virtual Route Pacing” on page 310. Session pacing can be used to prevent overrun of the buffers of the LUs at the session ends (refer to “Setting Pacing Values”). Virtual route pacing can be used to avoid buffer congestion in VTAM and NCPs. Both kinds of pacing use SNA-defined mechanisms to control the flow of data.

Application Program Pacing

For information on how to use the APPL definition statement to specify pacing for application programs, see “Application Program Pacing” on page 337.

Setting Pacing Values

You can affect how VTAM does virtual route pacing through the PATH definition statement or the VR pacing window size calculation module.

For information on setting pacing values, see “Setting Pacing Values” on page 339.

Operating VTAM

Operating VTAM is the task of starting and stopping VTAM, activating and deactivating resources, and monitoring the domain. Although the operator usually performs these tasks, the system programmer must define the procedures that the operator uses when doing these tasks. These procedures are usually written by the system programmer in a locally-produced document called a run book.

A run book is a list of procedures for operators to follow. For example, you might want to list the start options that operators should use when starting different hosts. Following is an example of what might be in a run book:

After you activate S013H:

```
VARY NET,ACT,ID=S013H1
VARY NET,ACT,ID=S013H2
VARY NET,ACT,ID=S013H3
VARY NET,ACT,ID=S013H4
VARY NET,ACT,ID=S013H5
VARY NET,ACT,ID=S013H6
```

Notice that this gives very specific information about your own network, such as the names of the nodes.

For further information on how to operate VTAM, VTAM functions that make operating VTAM easier, deciding on local procedures, and writing a run book, see “Operating VTAM” on page 341. For operators who are learning about operating VTAM, see Chapter 1, “Introducing an Operator to VTAM” in *VTAM Operation*.

Verifying a VTAM Network

This section suggests steps to follow to verify that your VTAM network functions properly. Before you perform these steps, make sure that the status of all resources in your network is inactive.

For specific information about how to perform these steps and for the syntax of operator commands, refer to *VTAM Operation*. If these steps do not yield the expected results, double-check the definition statements for the resources that are affected. If this does not solve the problem, refer to *VTAM Diagnosis*.

To verify that your VTAM network functions properly, do the following:

1. Start VTAM and the appropriate trace facility (such as **MVS** GTF or **VM** TRSOURCE).
2. Activate a channel-attachment major node, begin tracing, and activate lines and peripheral physical units.
3. Activate an NCP, begin tracing and intensive mode error recording, and verify that you can dump the NCP.
4. Start the NetView program, if it is installed.
5. Activate a switched major node, and begin tracing for the physical units.
6. Start the VTAM or host subsystem application programs.
7. Start VTAM traces, and activate the logical units at the peripheral physical unit.
8. Use DISPLAY commands to verify:
 - Peripheral terminals and links
 - Route structure
 - Sessions between application programs and logical units.
9. Activate and verify additional physical units and NCPs.
10. Simulate normal operation with all physical units.
11. Try backup and recovery procedures.
12. Halt VTAM.

Chapter 6. Getting Started with VTAM - Detail

This chapter provides additional detail on the VTAM implementation issues discussed in Chapter 5, “Getting Started with VTAM - Overview” on page 125. Use this chapter as a reference when reading Chapter 5.

Throughout this chapter, the term *file* represents the following:

- MVS** Partitioned data set member
- VM** File
- VSE** Book.

Application Programs

This section contains the following subsections:

- Naming an Application Program
- Authorizing Application Facilities
- Data Compression
- Security Facilities
- Logon Mode Parameters
- High Availability Using Extended Recovery Facility **MVS**
- Cryptography in an XRF Environment **MVS**
- Using User Variables (USERVAR)
- Generic Resources **MVS**
- LU 6.2 in an XRF Environment **MVS**
- Persistent LU-LU Sessions **MVS, VSE**
- VTAM Common Network Services **MVS, VM**
- Cross-Memory Application Program Interface (API) **MVS**
- Allocating Private Storage
- Communicating with Start-Stop Devices
- LU 6.2 Application Programs
- LU 6.2 Security
- LU 6.2 Sync Point Services
- Selective Termination of Idle LU 6.2 Sessions.

Naming an Application Program

You can specify two different names for an application program, one required and one optional. The required application program name is the one that you specify in the label field of the APPL definition statement. The optional application program name is the one that you specify on the optional ACBNAME operand of the APPL definition statement.

If an application program name is not specified on the APPLID operand of the ACB macroinstruction, VTAM uses a name supplied by your operating system. If the name supplied by your operating system does not match a name on the APPL definition statement, the application program cannot open its access method control block (ACB) and is therefore not identified to VTAM. The following indicates what name is used for each operating system:

- **MVS** If the application program is started by a job step that is a procedure invocation, the name is the procedure step name in the job control language

(JCL) for the application program. Otherwise, it is the job step name in the JCL.

- **VM** The name is the user ID of the virtual machine that the application program runs in.
- **VSE** The name is the job step name in the job control language (JCL) for the application program.

Notes:

1. **MVS** TSO/VTAM and logon manager are VTAM application programs and must be defined to VTAM. For the specific coding requirements for the APPL definition statement that defines TSO/VTAM, see Appendix A, “TSO/VTAM ” on page 611. For the specific coding requirements for the APPL definition statement that defines logon manager, see Appendix D, “Logon Manager ” on page 653.
2. **VM** VSCS is a VTAM application program and must be defined to VTAM. Refer to Appendix B, “VM/SNA Console Support” on page 627 for the specific coding requirements for the APPL definition statement that defines VSCS.

If you are using an application program that needs information about the environment in which it is running in, you can code macroinstructions in that application program to obtain information about the following:

- VTAM version and release levels
- VTAM component and feature identifiers
- Application program functions provided by your release of VTAM
- Application program’s network name
- Application program’s ACB name
- HOSTPU name
- Network ID for your network.

For information on application program pacing see “Application Program Pacing” on page 337.

Authorizing Application Facilities

You can use the following facilities to enhance your application program environment:

- Passing and validating logon requests
- Overriding dial number digits
- Acquiring LU sessions
- Enabling parallel sessions
- Authorizing privileged paths.

Passing and Validating Logon Requests: You can write a logon exit routine that passes VTAM session requests to another application program. This allows the application program to validate the logon before accepting it. Passing an LU session request to another application program queues the LU session request to that application program. A session is established after the original session has ended. The AUTH=PASS operand on the APPL definition statement permits the passing of session requests.

Overriding Dial Number Digits for Dial or Token-Ring Connections: In a dial-out operation, an application program can be authorized to override the dial number digits and other parameters specified in the switched major node for the device being contacted. The device can be PU types 1, 2, or 2.1. Normally, if these parameters do not match in XID (exchange identification), the connection is terminated. However, you can authorize an application to override this checking. Coding the AUTH=ASDP operand on the APPL definition statement allows the application to supply dial parameters to the model switched major node during session initiation.

Notes:

1. This can cause a security exposure.
2. If an active APPN connection exists, VTAM may attempt to use it rather than dialing a new connection.

Acquiring LU Sessions: An application program acquires LU sessions by issuing either an OPNDST macroinstruction with OPTCD=ACQUIRE or a SIMLOGON macroinstruction followed by an OPNDST macroinstruction with OPTCD=ACCEPT specified.

You can enable the acquiring of sessions by coding the AUTH=ACQ operand on the APPL definition statement.

Enabling Parallel Sessions: Certain session protocols can prevent an application program from performing more than one transaction at a time on a single session. Using the parallel sessions function, you can enable an application program to engage in several sessions at once with the same logical unit. Thus, the application program can perform multiple transactions at the same time.

If you want an application program to have parallel sessions, you must authorize it by coding PARSESS=YES on the APPL definition statement. APPC=YES overrides PARSESS=NO to allow parallel sessions. The application program can distinguish sessions between the same LUs through the proper use of the communication identifier (CID).

Authorizing Privileged Paths (MVS): Authorized path is a VTAM facility that enables an application program to specify that a data transfer operates in a privileged manner. Authorized path enables an authorized application program to run entirely under a system request block (SRB) after the application program's ACB has been opened. In most cases, this means a shorter path length through VTAM for RPL-related macroinstructions. An application program can be authorized by using the authorized program facility (APF) running in the supervisor state or by running in a key of 0 through 7. For a full description of authorized path, see "Authorized Path (MVS)" in *VTAM Programming*.

Data Compression

When using application programs in a multiple-domain environment, consider taking advantage of the data compression facility. This facility enables VTAM to compress the data for messages on selected LU-LU sessions. Data compression is transparent to application programs.

When using application programs in a single-domain environment, between application to application sessions in the same host, there is no benefit to compressing

data. In fact, extra processing cycles are expended. If the SLU is an application in the same host, VTAM builds the BIND without data compression even if data compression is indicated. Data compression is not allowed in these single-domain conditions.

Also, you should not use the data compression facility for VTAM in conjunction with any application program data compression. After the application program has compressed the data, the amount of additional compression would be minimal.

You can implement data compression for logical units in either an extended BIND (type 2.1, 4, or 5 physical unit) or non-extended BIND environment (type 1 or 2 physical unit). You cannot compress data for sessions with LU type 4 or LU type 7 in a non-extended BIND environment.

VSE Data compression uses data space storage.

Types of Compression

The use of data compression can be negotiated for each session through the BIND and BIND(RSP) request units (RUs). The session partners can negotiate whether to use data compression. If compression is to be used, the session partners can negotiate the type of compression. VTAM supports two different compression algorithms:

- Run length encoding (RLE) compression

This type of compression replaces strings of identical bytes with shorter encoded strings.

- **MVS, VSE** Dynamic dictionary-based compression (called adaptive compression in this book)

This type of compression is an adaptive dictionary-based compression algorithm similar to Lempel-Ziv. It is applicable to a wide range of data types. It uses tables that adapt dynamically to match the data being sent or received. Adaptive compression replaces the original data with a set of compression codes. Each set represents one or more bytes. Following are the variations of adaptive compression:

Small table This type uses 9-bit compression codes.

Medium table This type uses 10-bit compression codes.

Large table This type uses 12-bit compression codes.

With adaptive compression, each partner begins a session with an identical set of tables. The sender of the RU updates his send tables as data is compressed. The receiver of the RU makes identical updates to his receive tables as data is decompressed. This method keeps both ends of the session identical without exchanging table data between nodes.

By default, the send and receive tables are continually updated. You can speed up adaptive compression processing on outbound messages, through software only or with the help of hardware compression, by using the CPMIPS start option. By specifying a value between 1 and 99 (default value is 100), VTAM suspends and resumes updating of the send and receive tables whenever compression effectiveness exceeds or falls below a threshold corresponding to the CPMIPS start option value.

Thus, the CMPMIPS option allows you to balance the number of machine cycles needed with the effectiveness of compression for outbound messages. Higher values for CMPMIPS will likely increase both compression effectiveness and cycle usage, while lower CMPMIPS values will likely lower both compression effectiveness and cycle usage.

Implementing Data Compression

If you choose to use data compression, you can adjust your system's use of compression by using a combination of compression levels specified for VTAM and for application programs. The following levels of data compression are available:

0	No compression	
1	RLE	
2	Adaptive small table	MVS,VSE
3	Adaptive medium table	MVS,VSE
4	Adaptive large table	MVS,VSE

- The overall compression limit for a VTAM host can be:
 - Initialized through the CMPVTAM start option. For more information on the CMPVTAM start option, see “CMPVTAM” in the *VTAM Resource Definition Reference*.
 - Changed by the MODIFY COMPRESS command. (The compression levels of active sessions are not affected.) For more information on this command, see “MODIFY COMPRESS Command” in *VTAM Operation*.
 - Displayed through the DISPLAY VTAMOPTS command. For more information on this command, see “DISPLAY VTAMOPTS Command” in *VTAM Operation*.
- Input and output compression limits for an application program can be:
 - Initialized through the CMPAPPLI and CMPAPPLO operands on the APPL definition statement.
 - Changed by the MODIFY COMPRESS command. (The compression levels of active sessions are not affected.)
 - Displayed through the DISPLAY ID command.
- COMPRES is the compression-override operand on the MODEENT macroinstruction that specifies how the negotiation of compression should be handled. You can use COMPRES to override the BIND negotiation process and either require or prohibit compression. This operand allows data compression specifications to be associated with a logon mode name. For more information on this operand, see “COMPRES (MVS)” in the *VTAM Resource Definition Reference*.
- The DISPLAY SESSIONS command can indicate the compression levels in use for a given session and the average message length reductions achieved in both directions. For more information on this command, see “DISPLAY SESSIONS Command” in *VTAM Operation*.

Compression Level Negotiation

If the value specified on the COMPRES operand of the MODEENT macroinstruction allows negotiation, the compression levels for a session are negotiated through the BIND and BIND(RSP) RUs.

The data compression information is carried on a control vector X'66'. Control vector X'66' is used only for compression-level negotiation. The control vector is comprised of these subvectors:

X'80' is passed on a CDCINIT and a CINIT.

X'81' is passed on a BIND to carry the requested compression levels.

X'82' is passed on a BIND(RSP) to carry the actual compression levels.

Figure 41 illustrates an example of the negotiation process.

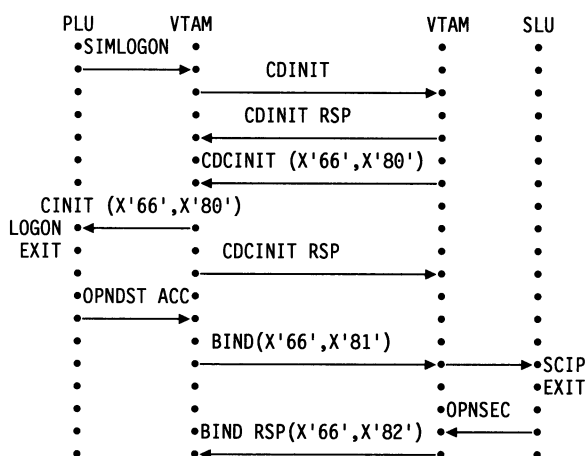


Figure 41. Example Session Flow for Compression-Level Negotiation

In the example illustrated in Figure 41, the SLU first requests a compression level on the CDCINIT. The PLU responds with requested levels specified on the BIND. The SLU responds with actual levels on the BIND(RSP).

Note: There are limited negotiation capabilities for sessions using a non-extended BIND. Bits 6 and 7 in byte 25 of the BIND are reserved for data compression, and are used to negotiate whether compression is acceptable or not. No level negotiation is possible. In the PLU-to-SLU direction, level 2 is the highest possible compression level (If VTAM owns the PLU, setting CMPVTAM = 1 further limits the PLU-to-SLU level to 1). In the SLU-to-PLU direction, level 1 is the highest possible compression level.

Negotiation When VTAM Owns the PLU: When VTAM owns the PLU for the session, VTAM has the option to request compression. VTAM does not request compression if any of the following conditions occur:

- The CMPVTAM start option is set to 0, which indicates no compression.
- The CMPAPPLI and CMPAPPLO operands on the APPL definition statement are set to 0, which indicates no compression.
- The SLU owner prohibits compression through bits in the control vector X'66'. Whether the control vector indicates that compression is prohibited or required is determined by the setting of the COMPRES operand of the MODEENT macroinstruction.

If none of the previous conditions occur, the PLU requests compression levels. The PLU-to-SLU compression level is the lower of the values set on the CMPVTAM start option or the CMPAPPLO operand on the APPL definition statement. The SLU-to-PLU compression level is the lower of the values set on the CMPVTAM start option or the CMPAPPLI operand on the APPL definition statement.

Negotiation When VTAM Owns the SLU: When VTAM owns the SLU for the session, the SLU can begin negotiation when it sends the control vector X'66'. The control vector indicates whether compression is prohibited or required. If compression is desired, the COMPRES operand on the MODEENT macroinstruction must indicate that compression is required.

When sending the BIND(RSP), the SLU negotiates no compression if any of the following occur:

- The BIND does not request compression.
- PROHIB or SYSTEM was specified in the control vector X'66'.
- The CMPVTAM start option is set to 0, which indicates no compression.

Note: When compression is negotiated from the SLU side, the CMPAPPLI and CMPAPPLO operands are not used.

If none of the previous conditions occur, the SLU requests compression levels. The PLU-to-SLU compression level is the lower of the values set on the CMPVTAM start option or the requested PLU-to-SLU compression level passed on the BIND. The SLU-to-PLU compression level is the lower of the values set on the CMPVTAM start option or the requested SLU-to-PLU compression level passed on the BIND.

Examples of Data Compression: The following are examples of how the values set for CMPVTAM, CMPAPPLI, and CMPAPPLO work together to determine the compression levels for a session. The examples are based on the configuration shown in Figure 42.

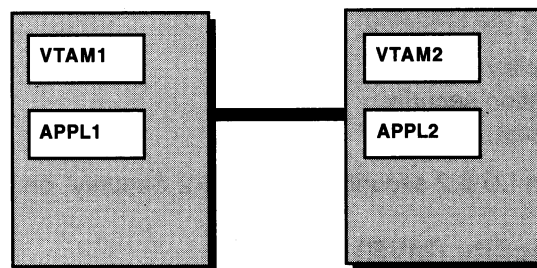


Figure 42. Data Compression Yield

For the examples, assume the values shown in Table 14. Also assume that COMPRES=REQD is specified on the logon modes used for the sessions.

Table 14. Compression Values for Example of Data Compression Yield

VTAM1	APPL1	APPL2	VTAM2
CMPVTAM = 4	CMPAPPLI = 4	CMPAPPLI = 4	CMPVTAM = 3
	CMPAPPLO = 1	CMPAPPLO = 2	

In the first example, APPL1 is the PLU and APPL2 is the SLU.

1. VTAM determines that the requested PLU-to-SLU level (which is the lower of the values specified on the CMPVTAM start option for VTAM1 or the CMPAPPLO operand for APPL1) is 1.
2. VTAM determines that the requested SLU-to-PLU level (which is the lower of the values specified on the CMPVTAM start option for VTAM1 or the CMPAPPLI operand for APPL1) is 4.
3. APPL1 passes the requested levels to APPL2 on the BIND.
4. VTAM compares the requested levels to VTAM2's CMPVTAM level. APPL2 determines that the actual PLU-to-SLU level for this session is 1 and the actual SLU-to-PLU level for this session is 3.
5. APPL2 returns the actual levels on the BIND(RSP).

In the second example, APPL2 is the PLU and APPL1 is the SLU.

1. VTAM determines that the requested PLU-to-SLU level (which is the lower of the values specified on the CMPVTAM start option for VTAM2 or the CMPAPPLO operand for APPL2) is 2.
2. VTAM determines that the requested SLU-to-PLU level (which is the lower of the values specified on the CMPVTAM start option for VTAM2 or the CMPAPPLI operand for APPL2) is 3.
3. APPL2 passes the requested levels to APPL1 on the BIND.
4. VTAM compares the requested levels to VTAM1's CMPVTAM level. APPL1 determines that the actual PLU-to-SLU level for this session is 2 and the actual SLU-to-PLU level for this session is 3.
5. APPL1 returns the actual levels on the BIND(RSP).

Security Facilities

VTAM provides a variety of security facilities that can be enabled for an application program, including the following:

- Encryption facility
- VTAM application security
- Confidential data.

For information on LU 6.2 security, see “LU 6.2 Security” on page 199.

Encryption Facility (MVS)

The encryption facility protects data passing over lines between network resources by permitting enciphering and deciphering of data for LU-LU sessions. Encryption is available for both LU 6.2 and non-LU 6.2 sessions. For an application program or peripheral node logical unit to have cryptographic sessions, the host processor must support cryptography.

You can use encryption for switched and non-switched LUs, but not for binary synchronous communication (BSC) or local non-SNA devices.

To encrypt data, you can use either of the following facilities:

- IBM Programmed Cryptographic Facility
- VTAM Integrated Cryptographic Service Facility.

To use the encryption facility:

1. Install and activate the IBM Programmed Cryptographic Facility, if using it. The ENCRYPTN start option enables you to start VTAM before activating the cryptographic product. The encryption facility provides two levels of cryptographic sessions: *selective* and *required*.

In *selective* cryptographic sessions, only data request units specified for encryption by the session ends are enciphered before transmission.

In *required* cryptographic sessions, all outbound data request units are enciphered and all inbound data request units are deciphered.

2. File cryptographic keys in the cryptographic key data set at the host processor before activating any LUs that are used in cryptographic sessions. For information about filing cryptographic keys, refer to Appendix E, "Cryptographic Keys" on page 663.
3. Code the ENCR operand on the LU and APPL definition statements to define the cryptographic capabilities of LUs and application programs. The ENCR operand on the MODEENT macroinstruction can be used to specify what type of cryptography an LU expects for a session. The coding of the ENCR operand on the APPL definition statement, the LU definition statement, and the MODEENT macroinstruction work together to determine which sessions will be encrypted. For information about the options you can specify on the ENCR operand, see "ENCR (MVS)" in the *VTAM Resource Definition Reference*.

Figure 43 illustrates an encryption scenario. If APPL1A initiates a session with LU1A, a session can be started with no encryption. However, if APPL1A initiates a session with LU1B, the session is encrypted. If APPL1B attempts to initiate a session with LU1A, the session fails. APPL1B requires encryption and LU1A does not support encryption. However, APPL1B can have an encrypted session with LU1B.

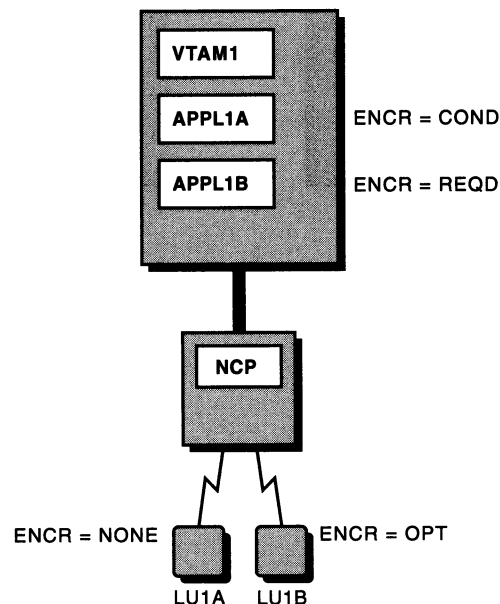


Figure 43. Encryption Facility in Single-Domain Environment

The cryptographic requirements of both session partners are checked when the session is established. A randomly generated session cryptography key, used to encipher and decipher data, is obtained for each cryptographic session.

The VTAM operator can use the MODIFY ENCR command to define the cryptographic capability of a logical unit, or change the capability to a higher level of cryptography than was previously specified. When VTAM tracing is in effect for a cryptographic session, the protected data in the VTAM buffers is neither displayed nor printed.

When implementing the encryption facility in more than two VTAMs, you need to ensure that intermediate VTAMs can support the encryption level you require.

In a subarea network, the session cryptography key can be encrypted on a host-by-host basis. With encryption on a host-by-host basis, cryptographic key data sets must be filed at each host that is involved in the LU-LU session initiation. If an LU-LU session is through an intermediate VTAM that does not support cryptography, that session cannot use encryption. For example, in Figure 44, VTAM2 must support cryptography if any sessions between resources in VTAM1's domain and VTAM3's domain are to be encrypted. Cryptographic key data sets must be filed at VTAM2, as well as VTAM1 and VTAM3. If VTAM2 does not support cryptography, then the following is true:

- APPL1A and LU3A cannot have a session.
- APPL3A and LU1A or LU1B cannot have a session.
- APPL1A and LU3B can have a session that is not encrypted.
- APPL3A and LU3A or LU3B can have encrypted sessions.
- APPL1A and APPL3A cannot have a session.

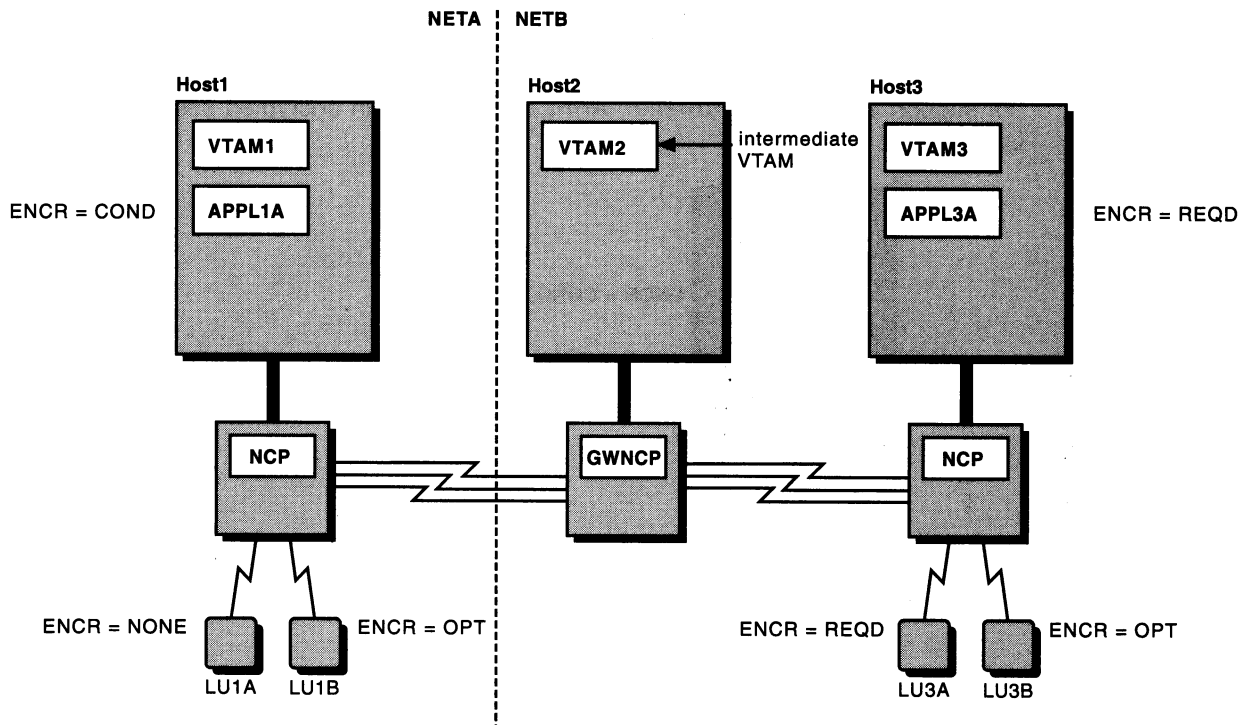


Figure 44. Encryption Facility in Multiple-Network Environment

VTAM also provides end-to-end cryptography support for subarea, APPN, and mixed subarea and APPN networks. End-to-end cryptography enables an LU-LU session to use encryption as long as all intermediate VTAMs are V3R2 or later and both end-point VTAMs are V4R1 or later. Instead of filing cryptographic key data sets at each host that an LU-LU session initiation request traverses, the data sets are required at only the end-point hosts. Defining cryptographic key data sets

between only end-point hosts eliminates the need to share cryptographic key information with other hosts. This can be important if, for example, the intermediate hosts are owned by a different company. Using end-to-end cryptography in Figure 44, cryptographic key data sets would need to be filed in only VTAM1 and VTAM3.

Defining cryptographic keys between only end-point hosts can still become burdensome, since many end-points can exist and the amount of key definition can get very large. Therefore, in an APPN network, cryptographic key data sets can be defined:

- Between a network node server and its end nodes, and
- Between network node servers.

Although more key definition is required at a network node server using this strategy, the overall number of cryptographic key data sets declines. End nodes are required to define only one key to the network node server (rather than a key for each node the end node is to communicate with). In addition to the keys for its served end nodes, network node servers are required to define only keys for other network node servers (rather than a key for each node its end nodes are to communicate with).

Note that both session end points do not have to utilize their network node servers for cryptography. That is, one end node can define a key to its server and the server can define a key for the destination end node (instead of defining a key for the destination end node's network node server).

An example of defining cryptographic keys using the network node server is given in Figure 45 on page 172. If VTAM1 files a cryptographic key data set with VTAM2 and VTAM2 files cryptographic key data sets with VTAM4, applications and LUs at VTAM1 can have cryptographic sessions with applications or LUs at VTAM4 without the need for VTAM1 and VTAM4 to exchange keys.

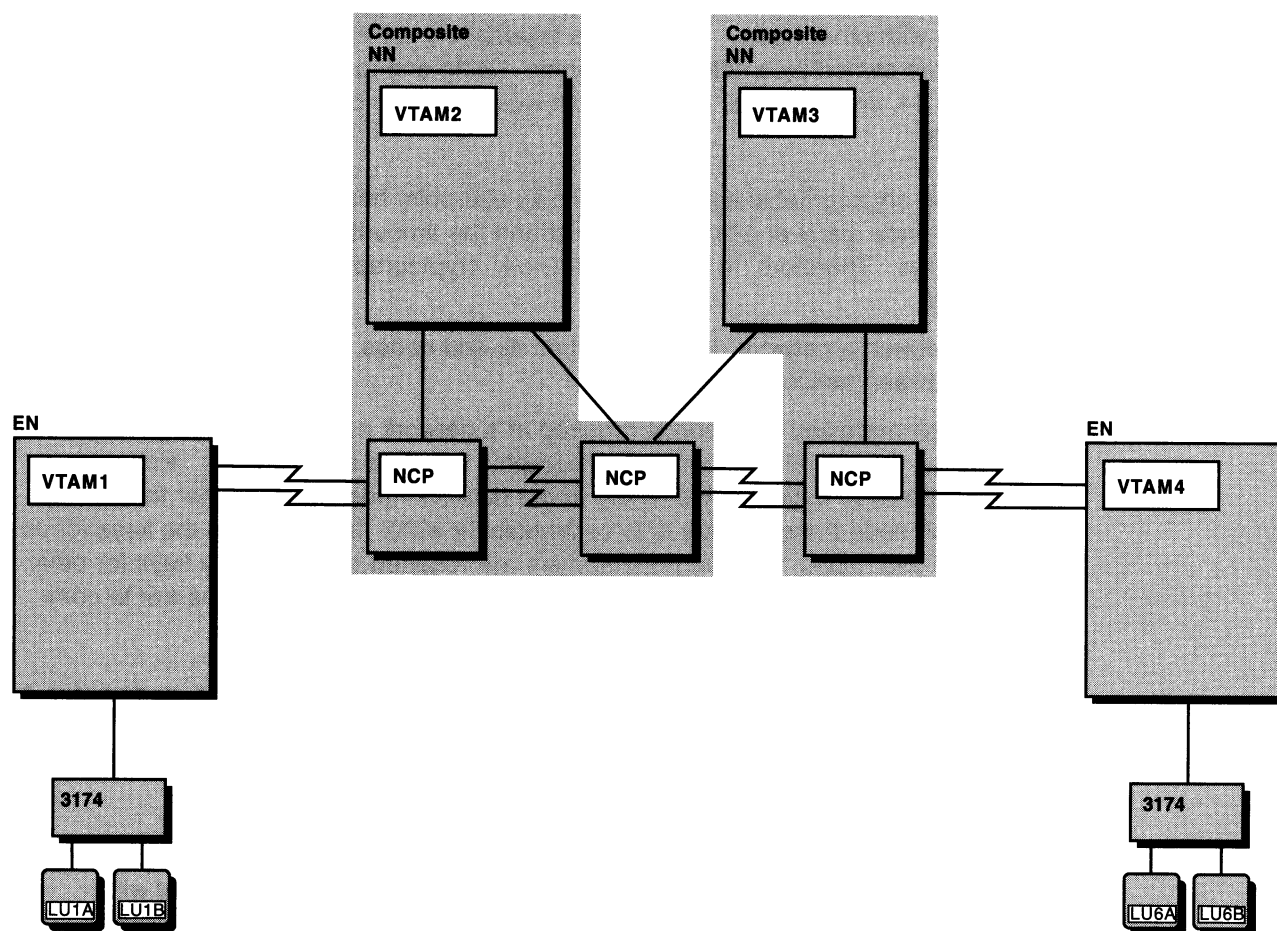


Figure 45. Encryption Facility in an APPN Environment

VTAM Application Security (MVS)

You can specify a password on the APPL definition statement and require the application program to specify both its APPL definition statement name and its password when it opens its access method control block (ACB). The authority of the application program to gain access to the network can be verified by comparing the password in the program's OPEN ACB macroinstruction to the password specified on the PRTCT operand of the APPL definition statement.

The VTAM application OPEN ACB security facility provides additional resource access control. To perform security processing, VTAM invokes a security management product (such as RACF) * through the security authorization facility for any application program that is not APF-authorized. The security management product determines whether an application program's access to the network is approved.

To enable the VTAM application OPEN ACB security facility, register the application program with a class of VTAMAPPL (CLASS=VTAMAPPL) in a security management product that is capable of controlling authorization for VTAM application program execution (such as RACF Version 1 Release 9). VTAM bypasses any password checking if the security management product provides the resource access control. The application program is authorized to access the network based on either the approval of the security management product or the user-specified password check.

The authority of the LU to initiate a session with an application program can also be verified by requiring that the LU include a password in its logon. This password can be verified by the session authorization function of a session management exit routine, a session authorization exit routine, the application program's logon exit routine, or the application program itself.

Confidential Data

When data is transmitted between an application program and a logical unit, it passes through VTAM buffers. These buffers are allocated from common page-fixed buffer pools when I/O is being performed and from pageable buffers in user-protected storage when data is queued. An application program that is transmitting or receiving confidential data can have fixed buffers cleared if PROC=CONFTXT is specified in the appropriate node initialization block (NIB). If you use this option, only the name of the application program, the name of the logical unit, and the direction of the data flow are included in the trace records; confidential data is not included. A buffer trace of non-confidential data includes the data.

Following are two ways of protecting application program data:

- The application program can ensure that confidential data in VTAM buffers within the host of the application program is cleared after the data is sent to the NCP or is moved into the application program address space. The application program does this by using the CONFTXT option in the node initialization block (NIB) associated with the session.
- An LU 6.2 application program, which provides a LOGON or SCIP exit routine to accept LU 6.2 sessions, can also use a confidential test option. You can code the CONFTXT=YES operand on the APPCCMD macroinstruction used to accept an LU 6.2 session (APPCCMD CONTROL=OPRCNTL,QUALIFY=ACTSESS). If you code CONFTXT=YES, VTAM clears its buffers that have been used to hold application program data prior to returning them to VTAM buffer pools for reuse.

Note: TSO and VSCS also provide a CONFTXT option that allows you to protect confidential data. For information on CONFTXT and TSO, see "Security" on page 621. See "Protecting Confidential Data" on page 633 for information on CONFTXT and VSCS.

Logon Mode Parameters

If you have an application program that acts as the secondary end of the session, you might need to create new entries or supplementary tables, as follows:

- If the application program is the secondary logical unit, you might want to specify session protocols. You can direct VTAM to the proper session parameters by specifying the logon mode table entry (through the DLOGMOD operand) and the logon mode table (through the MODETAB operand) on the program's APPL definition statement.
- If an application program is intended to be in session with specific devices or application programs and it requires certain protocols, you might have to create new logon mode table entries associated with the devices.

Both an APPN class of service and a subarea class of service, specified in the logon mode entry, can be associated with each session. The APPN class of service determines the route that is used for APPN links, and the subarea class of service determines the virtual route that is used for a session. Therefore, you

should understand the class-of-service requirements for your application programs. Considerations such as the importance of the function provided by the application program, the nature of the data exchanged with the logical units, and the expected response times contribute to determining the class of service needed for sessions with the application program. The class-of-service names must then be specified in the logon mode table entries used for LU-LU sessions. For more information about logon mode entries, see “Selecting Session Parameters for the Logon Mode Table” on page 158.

High Availability Using Extended Recovery Facility (MVS)

You can identify an application program as capable of operating in an extended recovery facility (XRF) environment. Two IBM application programs, Information Management System (IMS) and Customer Information Control System (CICS), support an XRF environment.

Note: The USERVAR facility can be used under the MVS or VM operating systems. However, the primary application programs that use the USERVAR facility are CICS and IMS in an extended recovery facility (XRF) environment, an environment that operates only under the MVS operating system.

XRF provides an alternate application program when certain host components fail. It reduces the impact of planned outages or outages caused by failures in VTAM, MVS, IMS, CICS, or the host processor on selected users of IMS, CICS, or other application programs that use XRF functions. The types of failures that cause the takeover of an XRF session are determined by how IMS, CICS, or the application program using XRF is defined. XRF might not reduce the impact of failures in the NCP or in telecommunication lines, depending on which NCP or line fails. However, the XRF function in CICS or IMS does not automatically perform a switch to the alternate application program under these conditions. Therefore, either the operator must switch from the active to the alternate system, or you can use NetView automation to perform the switch.

To enable high availability, use the HAVAIL operand on the APPL definition statement.

VTAM, in conjunction with the application program, establishes a primary session from the currently active application program and a backup session from the alternate application program to the terminal. The NCP for a terminal in an XRF environment maintains two sessions through two VTAMs: one to a primary XRF application program and one backup session to the alternate XRF application program. The XRF application programs, in conjunction with VTAM and NCP, switch to the backup session when some component fails. The active and alternate application programs must be in the same network. However, a terminal participating in an XRF session can be in a different network than either of the application programs.

One application program acts as the active application program, and the other acts as the alternate in case the active application program or another component in the active host fails. Both application programs are identified by a single name that is associated with the currently active application program.

Although one VTAM in a single-processor environment can support both a primary and a backup session, this configuration does not provide the availability characteristics usually associated with an XRF environment.

XRF is also supported for application programs that use LU 6.2 sessions to communicate with other applications or logical units in peripheral nodes attached to an NCP. For further information, see “LU 6.2 in an XRF Environment ” on page 193.

Cryptography in an XRF Environment (MVS)

XRF application programs can have cryptographic sessions with other logical units. Cryptography protects data passing over lines by permitting enciphering and deciphering of data for LU-LU sessions. The XRF application program must reside in a host that has the IBM Programmed Cryptographic Facility or the VTAM Integrated Cryptographic Service Facility program product installed and active.

The IBM Programmed Cryptographic Facility must be started before starting VTAM. It also must be started before you activate an external CDRM or CP for which CROSS statements have been defined in the cryptographic key data sets. This is not necessary for the VTAM Integrated Cryptographic Service Facility.

The XRF application program can specify that cryptography is either selective (specified by the session end) or required (session ends must support cryptography).

To enable this facility, code the following:

- ENCR operand on the APPL definition statement for the XRF application program
- ENCR operand on the LU definition statement for any logical units that will be session ends
- Cryptographic keys in the cryptographic key data set.

For more information about coding the ENCR operand, see “Encryption Facility ” on page 168. For more information about coding cryptographic keys, see Appendix E, “Cryptographic Keys ” on page 663.

Using User Variables (USERVAR)

The VTAM user variable (USERVAR) maps a generic application program name used in a terminal logon to the name of a specific application program, based on the value of the USERVAR. This function is used by IMS and CICS XRF to map user logons to the IMS or CICS application program that is currently active.

You can also use USERVARs in other ways, such as to facilitate migration from one application program release to another. The VTAM user variable allows the following:

- Your installation to use the USERVAR function without having to rely upon NetView command lists or manual procedures to propagate USERVAR updates. VTAM communicates USERVAR values across domains in the same or different networks.
- Your application programs to use USERVAR names across VTAM's application program interface (API) in place of logical unit names without requiring code changes.
- Your terminals to take advantage of USERVAR translation without your having to code interpret tables for them.

The VTAM application program interface (API) allows application programs to use USERVAR names across the API in place of LU names. Because USERVAR can be used across the API, many application programs that participate in sessions with an XRF-capable application program (such as IMS or CICS) can remain unchanged when USERVARs are used.

Also, because VTAM determines whether a name refers to an LU or to a USERVAR and performs the appropriate translation automatically, you do not have to code an interpret table to specify that a name used in a particular logon is actually a USERVAR.

Note: The generic name used by USERVAR should not be used by ALIAS. An alias CDRSC should not exist in a host where USERVAR mapping is used unless the host is in the same network as the real resources.

Application Workload Balancing with USERVAR

A group of applications can be given a common or “generic” name. VTAM uses this generic name, known as a USERVAR, to associate a logon request with the currently active member of this group. This capability can be used for workload balancing and preparing for the planned takedown of a host. USERVAR values can be managed by the user as well as by VTAM.

For example, assume that the USERVAR, IMS, has been defined, and that there are two IMS application programs: IMS1 and IMS2. The generic name (USERVAR) can be assigned the value of either IMS1 or IMS2, based on which system is active. If the USERVAR for the IMS application programs is set to IMS1, for instance, then any session requested with IMS is routed to IMS1. If IMS is set to IMS2, then new sessions are routed to IMS2. Therefore, if the workload going to IMS1 is very heavy, the IMS USERVAR can be set to IMS2 and subsequent sessions are routed to IMS2.

The following commands display and modify the VTAM application names associated with USERVARs:

- DISPLAY USERVAR
- MODIFY USERVAR
- DISPLAY ID
- DISPLAY SESSIONS.

For further information on these commands, see *VTAM Operation*.

Classes of USERVARs

VTAM supports the following two classes of USERVARs:

- User-managed USERVAR
- Automatic (or VTAM-managed) USERVAR.

User-managed USERVAR

is explicitly set using the MODIFY USERVAR operator command. The MODIFY USERVAR command can be issued either manually by a human operator or automatically by a NetView command list or a VTAM application program (as is done by IMS or CICS for XRF).

VTAM does not attempt to change the value of a USERVAR that has been set explicitly using the MODIFY USERVAR operator command, so existing command lists and procedures for propagating USERVAR changes across domains in a network should continue to work as is.

However, a MODIFY USERVAR command can be used to delete a user-managed USERVAR to subsequently allow that USERVAR name to be managed by VTAM.

At least one host in the network must have a user-managed USERVAR from which other hosts can obtain the USERVAR value and type. In XRF situations, IMS and CICS both set the USERVAR value at the site of the XRF active application program.

Notes:

1. For XRF application programs, VTAM deletes user-managed USERVARs when the application program terminates or enters a specific state, such as CONCT or INACT.
2. When a VTAM in an APPN network (end node or network node) receives a request containing a USERVAR name, the user-managed USERVAR must be defined on the same node as the real resource. Therefore, when the real resource is on an end node, define the USERVAR at the end node and not at its network node server.
3. When an interchange node (in a combined APPN and subarea network) receives a request from the APPN network containing a USERVAR name, the real resource must be located on the interchange node or through an adjacent SSCP attached to the interchange node (that is, not in the APPN network attached to the interchange node).

Automatic (or VTAM-managed) USERVAR

is created automatically by VTAM in one domain as a copy of a user-managed USERVAR of the same name in another domain. You do not have to do anything to allow VTAM to create automatic USERVARs. When a VTAM SSCP sends a cross-domain session-initiation request to another domain and receives a reply indicating that the name specified in the original session request is really a USERVAR with a particular value and not the name of a logical unit, the SSCP creates an automatic (VTAM-managed) USERVAR to save the information. The cross-domain session request is then resent using the real LU name, which was passed back in the original reply.

Types of USERVARs

Changes to the value of a USERVAR are not propagated by VTAM at the time of the change. Instead, when a domain using a USERVAR to establish a session encounters a specific USERVAR for the first time, it uses an adjacent SSCP table to perform a cross-domain search to obtain the USERVAR's value. Subsequent attempts to use the same USERVAR can result in a repeat of the cross-domain search to update the USERVAR's value, depending on the value of the TYPE operand. The TYPE operand is specified on the MODIFY USERVAR command when the USERVAR is created. The TYPE operand is copied, along with the USERVAR's value, when an automatic USERVAR is created in another domain. Following are the types of USERVARs:

Static USERVAR

The value is assumed not to change. It is saved as an automatic USERVAR upon initial use but is not updated automatically using the cross-domain search process (unless the existing copy of the automatic USERVAR is explicitly deleted by the operator).

Dynamic USERVAR

The value is expected to remain relatively stable but change occasionally (for example, when an XRF takeover occurs). To establish a cross-domain session to an application program identified by a dynamic USERVAR, VTAM must recheck the USERVAR's value after every abnormal session termination involving the application program that it references.

DYNAMIC is the default value for the MODIFY USERVAR command and is recommended for use within XRF complexes.

Volatile USERVAR

The value is expected to change often. Rather than creating an automatic copy of the USERVAR, a remote VTAM repeats the cross-domain search to re-establish the USERVAR's current value each time it is referenced to establish a session. Due to the additional search flow that is necessary for each session establishment, use of volatile USERVARs might have a substantial impact on the time required for session establishments.

Note: If a USERVAR value is allowed to vary over multiple active application programs, LU 6.2 communication works only for a recovery environment. See "LU 6.2 in an XRF Environment" on page 193.

Processing USERVARs

Assume that an XRF application program, such as CICS, runs on Systems A and B of the sample network shown in Figure 46. CICSA is the specific VTAM APPLID of the CICS running on System A, and CICSB is the APPLID of the CICS running on System B. The terminal network is owned by the VTAM in System C (this is a communication management configuration, or CMC).

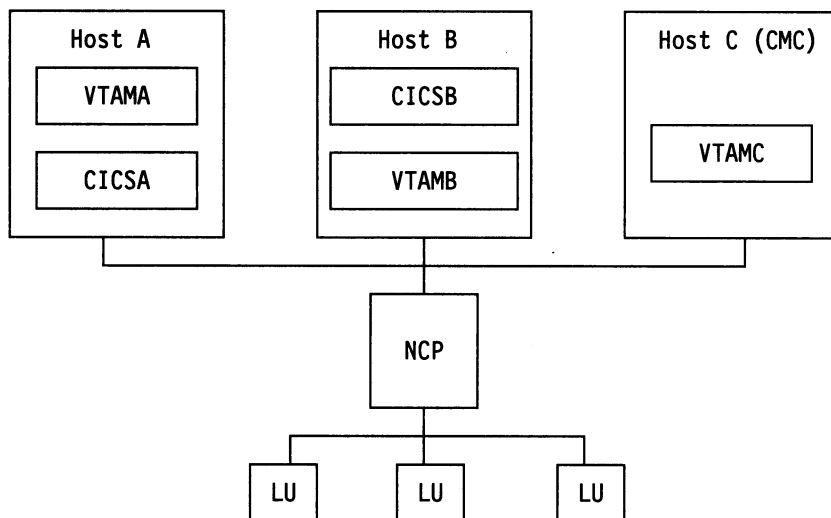


Figure 46. Sample Extended Recovery Facility Network

1. When CICS is started as an active CICS application program, it issues an F NET,USERVAR,ID=CICS,VALUE=CICS command, requesting that VTAMA create a dynamic (by default), user-managed USERVAR with a value of CICS. Because CICSB is started as an XRF alternate, it does not request that VTAMB create a USERVAR.

2. Logon requests from a terminal are processed by VTAMC, which is the controlling VTAM for the terminal network. The first time a terminal attempts to log on to generic application name CICS, VTAMC processes the logon request as follows:
 - Because VTAMC does not recognize the name CICS yet, it treats the request like any other cross-domain logon request and sends a cross-domain session request to other VTAMs listed in an ADJSSCP table looking for CICS. At this point, VTAMC does not know whether CICS is the name of a USERVAR or a real VTAM APPLID.
 - When VTAMA receives the session request, CICS is identified as the name of one of its own user-managed USERVARs. It replies to VTAMC indicating that CICS is really a dynamic USERVAR whose value is currently CICSA.
 - VTAMC uses the information it received from VTAMA to establish a cross-domain session with the application program by sending another session request using that specific application name, CICSA. In addition, VTAMC creates an automatic USERVAR to save the USERVAR information for future use.
3. For subsequent terminal logons to CICS, VTAMC uses the information it saved previously in the automatic USERVAR to translate the requests immediately to CICSA. VTAMC continues to translate CICS requests directly to CICSA without repeating the cross-domain search process because CICS is a dynamic USERVAR. This process continues until either an XRF takeover occurs or VTAMC discovers that it is otherwise unable to establish sessions with CICSA.
4. If CICSA fails and CICSB takes over and becomes the new active application program, CICSB issues an `F NET,USERVAR,ID=CICS,VALUE=CICSB` command, requesting that VTAMB create a new user-managed USERVAR with the value CICSB. This means that VTAMB is now the VTAM to identify session requests searching for CICS. VTAMB then responds, indicating CICSB is the current value of the USERVAR.

Each VTAM, which was notified that its XRF sessions were switched, deletes any automatic USERVARs that were used to create those sessions. This means that the next logon for CICS does not find the USERVAR and goes through the cross-domain search process (step 2) to determine the new value of CICS. In addition, the VTAM associated with the failing application program (VTAMA) deletes its user-managed USERVAR for CICS and prevents any further logons to CICSA to ensure that logons do not get directed to the failing application program.

Note: If a resource submits a network-qualified session request with a generic resource name, USERVAR processing occurs in the domain of the resource if:

- The NETID specified on the session request is equal to the NETID of the USERVAR's value, or
- The NETID specified on the session request is equal to the NETID of the host and the USERVAR value is not network-qualified.

USERVAR Propagation and Routing

NetView command lists are not required to propagate USERVAR modifications to a host in which VTAM can manage the USERVAR values; however, existing NetView USERVAR propagation command lists continue to work as they did before.

Note: VTAM can manage USERVAR values at a given host only if it and all hosts on the session establishment path have VTAM V3R2 or later and have installed the appropriate PTF for VTAM V3R2 that provides this USERVAR support.

VTAM searches for the value of a USERVAR through ADJSSCP table routing for a cross-domain session request. Therefore, if you have not already done so, you might need to define adjacent SSCP tables for searches to find the VTAM having the user-managed USERVAR. For information about this, see “Adjacent SSCPs” on page 398.

Generally, VTAM does not change user-managed USERVARs; they are viewed as being the responsibility of the user (or application program) that created them. However, in XRF situations, VTAM reduces the problem of session requests going to a failing XRF-active application program by deleting the user-managed USERVAR that references the failing application. VTAM deletes the user-managed USERVAR only at the site of the failing application program. If you have created a user-managed USERVAR at another host in the network, you need to make sure that USERVAR is deleted or that it continues to direct session requests to the failing application program.

For the VTAM having a user-managed USERVAR to respond to a session initiation request using that USERVAR, VTAM must know certain information about the LU that its value references (for example, the name of the SSCP that owns the LU). The necessary information is available if the referenced LU is owned by the host having the user-managed USERVAR, or if a CDRSC exists for the LU on that host. Otherwise, VTAM responds to the session initiation request as though the name were not found to allow the ADJSSCP routing to continue.

Defining Your Network with USERVARs

When you define your network resources for XRF, you should consider the following:

- If you allow dynamic CDRSCs and trial and error rerouting, you do not need to predefine a CDRSC for the USERVAR name or CDRSCs for the active and backup application program.

Note: Referencing dynamic CDRSCs with USERVARs can cause situations in which session initiations fail intermittently, because dynamically defined CDRSCs can be deactivated and deleted by VTAM, depending on the dynamic CDRSC timer value (CDRSCTI) and whether there are any active sessions.

- If you do not allow dynamic CDRSCs to be created in the network, you need to predefine a CDRSC for the USERVAR name, a CDRSC for the active application program, and a CDRSC for the backup application program. If you do predefine a CDRSC with the USERVAR name, you should not specify the cross-domain resource manager (CDRM) name because the USERVAR may not always be managed from the same SSCP. However, you should define the CDRSC under the network in which the USERVAR resides to reduce the amount of time for the ADJSSCP search process.

The following applies regardless of whether you use USERVARs with XRF:

- You can use USS tables, interpret tables, or both with USERVAR translation; the USS and interpret processing is done first and USERVAR translation is done using the output of those processes.
- You are not required to define interpret table entries for terminals to use the USERVAR function. If you have already done so, however, you do not have to remove them.
- If the application is cross-domain, the USERVAR name must be defined as a CDRSC using either static or dynamic definitions and must be unique within the network.

Dynamic USERVAR Update Session Failure

Whenever an abnormal session termination occurs for a session that was initiated using a USERVAR, VTAM deletes the USERVAR if it is VTAM-managed and dynamic. When the next session request arrives at any host other than the XRF primary host, VTAM repeats the search for the USERVAR and saves the new current value in an automatic USERVAR.

The following kinds of session termination are considered normal because they do not cause the resetting of a dynamic USERVAR:

- Requests for conditional or unconditional termination, such as the following:
 - Operator issued the VARY NET,TERM,TYPE=CONDIUNCOND command
 - SLU application program issued the TERMSESS OPTCD=CONDIUNCOND command
 - Terminal issued (using USS) the LOGOFF TYPE=CONDIUNCOND command
- Session setup failures due to the following sense codes:
 - 08210002
 - 088A0003
 - 088A0004
- LU-LU session terminations (UNBIND) for the following reasons:
 - Normal end of session (TYPE=X'01')
 - CLSDST PASS (TYPE=X'02')
 - Invalid session parameters (TYPE=X'06')
 - Other UNBINDs between X'01' and X'06'
 - Format or protocol error (TYPE=X'FE').

A session termination for any reason other than those reasons described previously is handled as an abnormal session termination and causes the deletion of any VTAM-managed dynamic USERVAR associated with that session. Following are examples of abnormal terminations:

- Requests for forced termination which, like requests for conditional or unconditional termination, can come from operators, SLU application programs, or terminals.
- LU-LU session terminations, including the following:
 - VR INOP (TYPE=X'07')
 - Route extension INOP (TYPE=X'08')
 - LU failure (TYPE=X'0C')

- Cleanup termination, such as V NET,INACT (TYPE=X'0F')
 - Gateway node cleanup (TYPE=X'11').
- Session setup failures other than those described preceding (such as failure to find the destination LU after using a dynamic USERVAR to determine the application program name). In such cases, VTAM deletes the dynamic USERVAR and retries the session initiation using the USERVAR name to determine the current value of the user-managed USERVAR.

Whenever the active application program in an XRF complex fails, notification of the session failures is sent to the SSCPs of the session partners, thereby causing them to reset their dynamic USERVARs. In the host of the XRF active application program, VTAM deletes all USERVARs that reference the failing application program to prevent future USERVAR searches from using a USERVAR that references the failing application program.

If a backup session already exists, an alternate application program sends a switch request to the NCP to takeover the sessions. The XRF takeover can be initiated before VTAM even knows that the active application program is failing. VTAM's first awareness of such a takeover is the receipt of a session ended notification having a specific session termination reason code (X'13'), which indicates that the primary session terminated due to an XRF switch. At a remote SSCP, such session terminations are handled as abnormal session terminations. At the SSCP of the failing active application program, however, the SSCP also does the following:

1. Marks the application program as “not enabled for sessions” to prevent any further session establishments
2. Deletes the USERVAR that is associated with the session that is ending.

Generic Resources (MVS)

Whereas a USERVAR is assigned to only one resource at a time, the generic resources function allows the assignment of a generic resource name to a group of active application programs that all provide the same function. The generic resource name is assigned to multiple active application programs simultaneously, and VTAM automatically distributes sessions among these application programs rather than assigning all sessions to a single resource. Thus, session workloads in the network are balanced. Session distribution is transparent to end users; an LU initiates a logon request using the generic resource name and need not be aware of which particular application program is actually providing the function.

The generic resources function also increases application program availability, as each active application program that uses a given generic resource name (a generic resource member) can backup other generic resource members. Thus, no single application program is critical to resource availability. When a generic resource member fails, an LU can reinitiate its session using the same generic resource name. VTAM resolves the session initiation to one of the other generic resource members. Because the user is unaware of which generic resource member is providing the function, the user is less affected by the failure of any single generic resource member.

In the same way, an application program can be added as a generic resource member and accessed using the same generic resource name. Because the user is unaware of which generic resource member is providing the function, an applica-

tion program can be added as a generic resource member with little or no impact to the application user. At the same time, the additional application program provides an immediate improvement in performance and availability since the workload is now shared by an additional application program.

Generic Resources Requirements

Any VTAM application program running on a VTAM APPN node that is connected to an MVS coupling facility structure named ISTGENERIC can be known by a generic resource name. All VTAM nodes having access to the common coupling facility structure ISTGENERIC, along with the ISTGENERIC structure itself, make up a generic resources configuration. Following are the requirements for implementing a generic resources configuration:

- A generic resources configuration must be part of a sysplex environment.
A sysplex is a set of MVS systems communicating and cooperating with each other, through certain multisystem hardware components and software services, to process customer workloads. For more information about sysplex, see *MVS/ESA Planning: Sysplex Management*.
- All VTAMs in the sysplex environment must be running under MVS/ESA V5R1 or a subsequent release of MVS that supports coupling facility services.
- An MVS coupling facility is required to provide the hardware and software that supports high-speed shared storage across multiple MVS systems and provide multiple copies or images of the MVS operating system to process work concurrently.
- There must be an active coupling facility resource management (CFRM) policy for the sysplex environment.

A CFRM policy defines how an installation will manage its coupling facility resources, which may be shared among multiple subsystems. For more information about how to create, activate, and make changes to the CFRM policy, see *MVS/ESA Planning: Sysplex Management*.

- The structure ISTGENERIC must be defined in the active CFRM policy for the sysplex.

All VTAMs in a generic resource configuration must be connected to an MVS coupling facility in which the structure, ISTGENERIC, may be allocated. VTAM connects to a coupling facility during VTAM initialization.

The coupling facility structure, ISTGENERIC, stores the data needed by VTAM to implement the generic resources function. This data includes mappings that show:

- For each generic resource name, a list that identifies the generic resource members currently using that name. This is called a generic resource mapping.
- For each LU in session with one or more generic resource members, the application program network name and the generic resource name for each generic resource member. This is called a partner LU mapping.

In defining ISTGENERIC in a CFRM policy, you specify the amount of coupling facility storage which should be allocated on behalf of this structure. For information on determining the amount of storage necessary, see “Defining a Structure Size for ISTGENERIC” on page 191.

- VTAM Version 4 Release 2 is required for all VTAMs in a generic resources configuration, and all VTAMs must be defined as network nodes or end nodes.

Generic Resource Mapping

A map of the application programs that are members of each generic resource is kept in ISTGENERIC. When an LU initiates a session using a generic resource name, VTAM establishes the session with one of the members that is mapped to that name.

Figure 47 shows how the mapping identifies the application programs that are using the different generic resource names. In this example, APPLA, APPLB, and APPLC are represented to the network as JOHN. APPLD is using the generic resource name SEAN.

MVS Sysplex

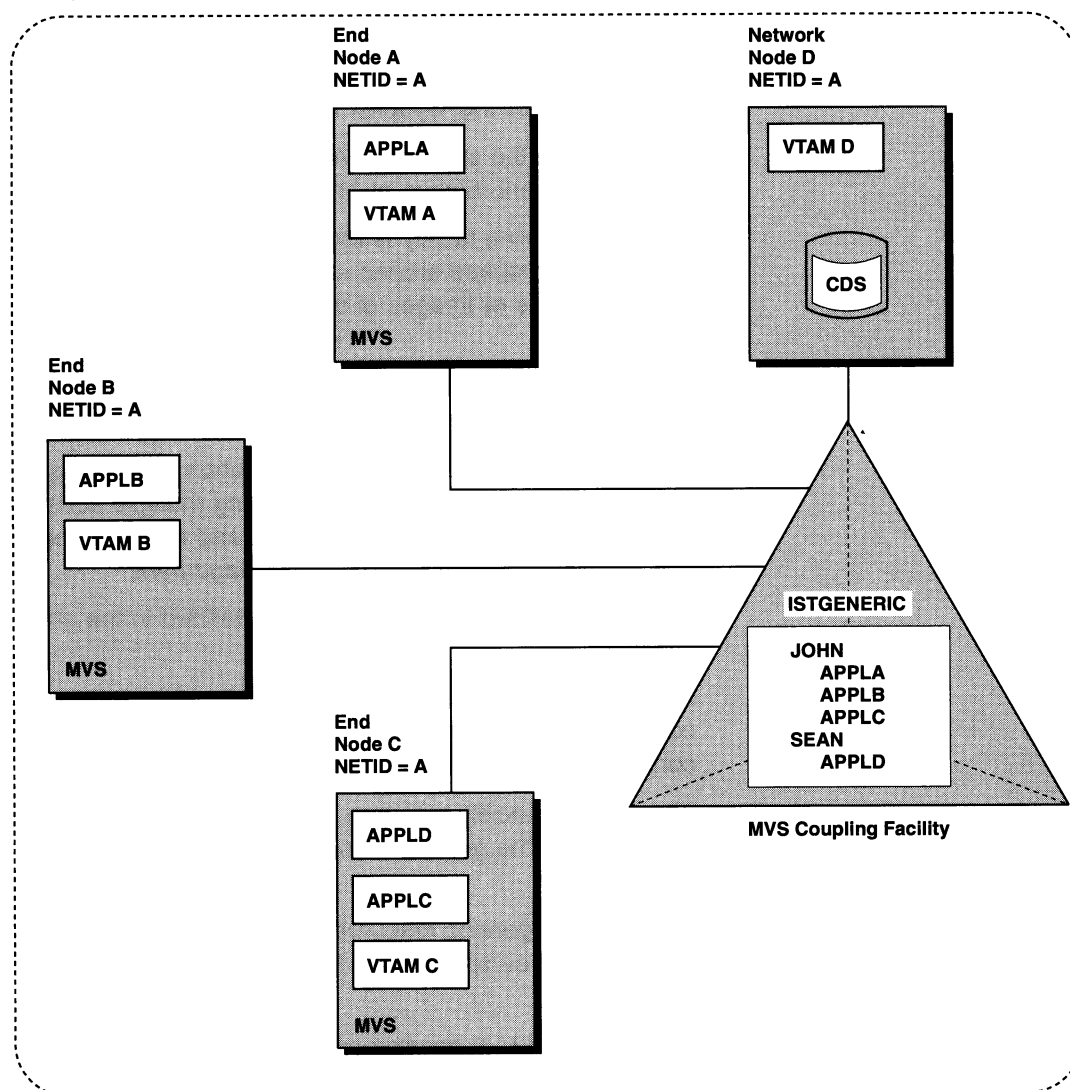


Figure 47. Generic Resource Mapping

Partner LU Mapping

ISTGENERIC also contains a mapping that identifies the partner LUs for each generic resource member. When the first session is established between an LU and a generic resource member, VTAM creates a mapping for that LU. This mapping, or affinity, identifies the name of the LU, the application program network name of the generic resource member, and the generic resource name. When subsequent parallel sessions are established using the same generic resource name, VTAM uses this mapping to ensure the session is established with the same generic resource member.

Figure 48 on page 186 shows partner LU mapping information. In this example, LU5 is in session with APPLB using the generic resource name JOHN and is in session with APPLD using the generic resource name SEAN. LU3 is in session with APPLA using the name JOHN.

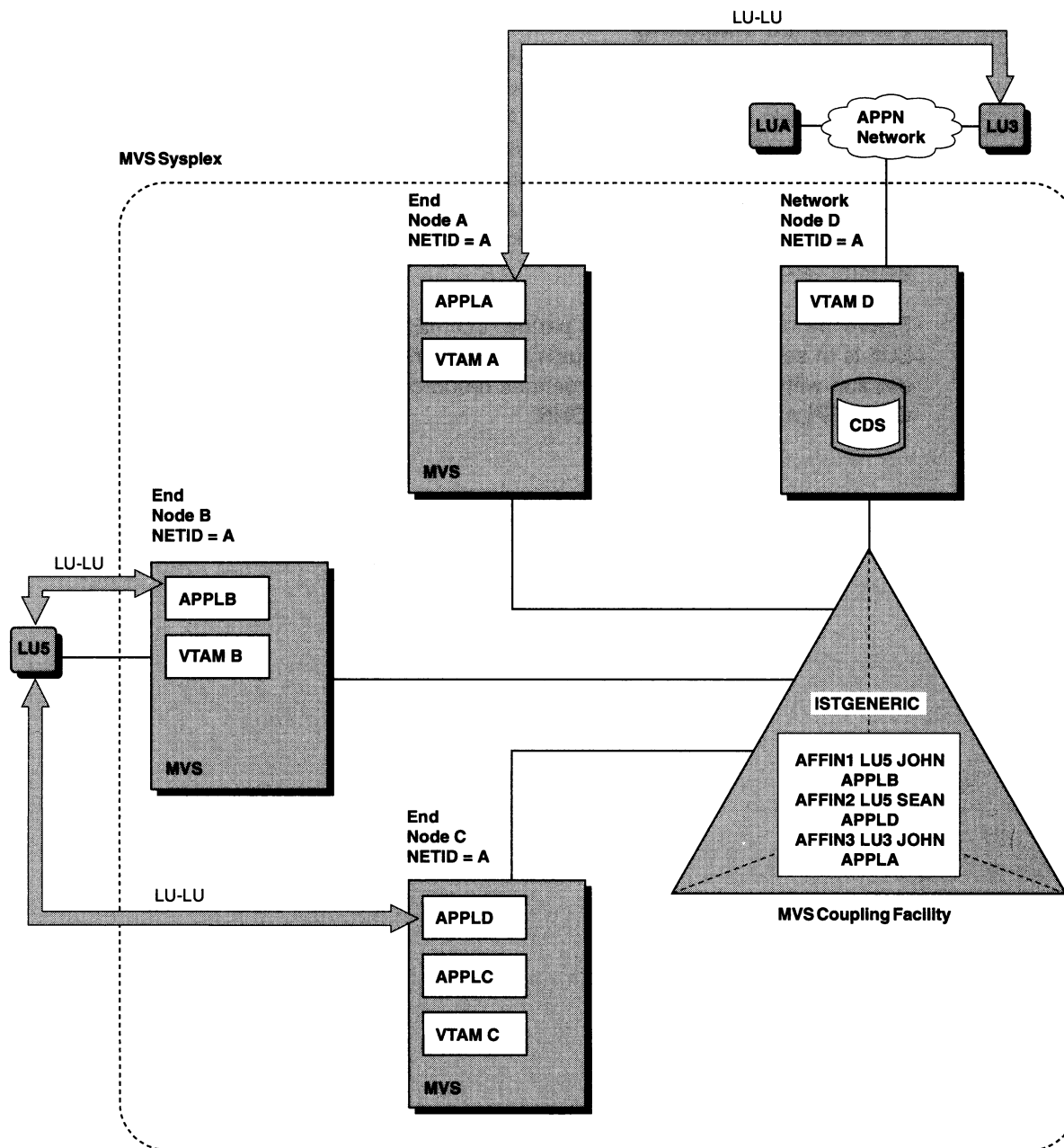


Figure 48. Partner LU Mapping

When LUA initiates a session with JOHN, VTAM determines whether LUA has an affinity with a generic resource member for JOHN. Because LUA does not have an affinity with any generic resource member using the name JOHN, VTAM establishes the session with the generic resource member having the least number of LU-LU sessions (in this example, APPLC).

Figure 49 on page 187 shows the affinity that is created after LUA establishes a session with JOHN. VTAM resolves the session to APPLC; LUA is only aware of having established a session with JOHN. All subsequent parallel sessions between

LUA and JOHN resolve to the same generic resource member, APPLC, regardless of session count.

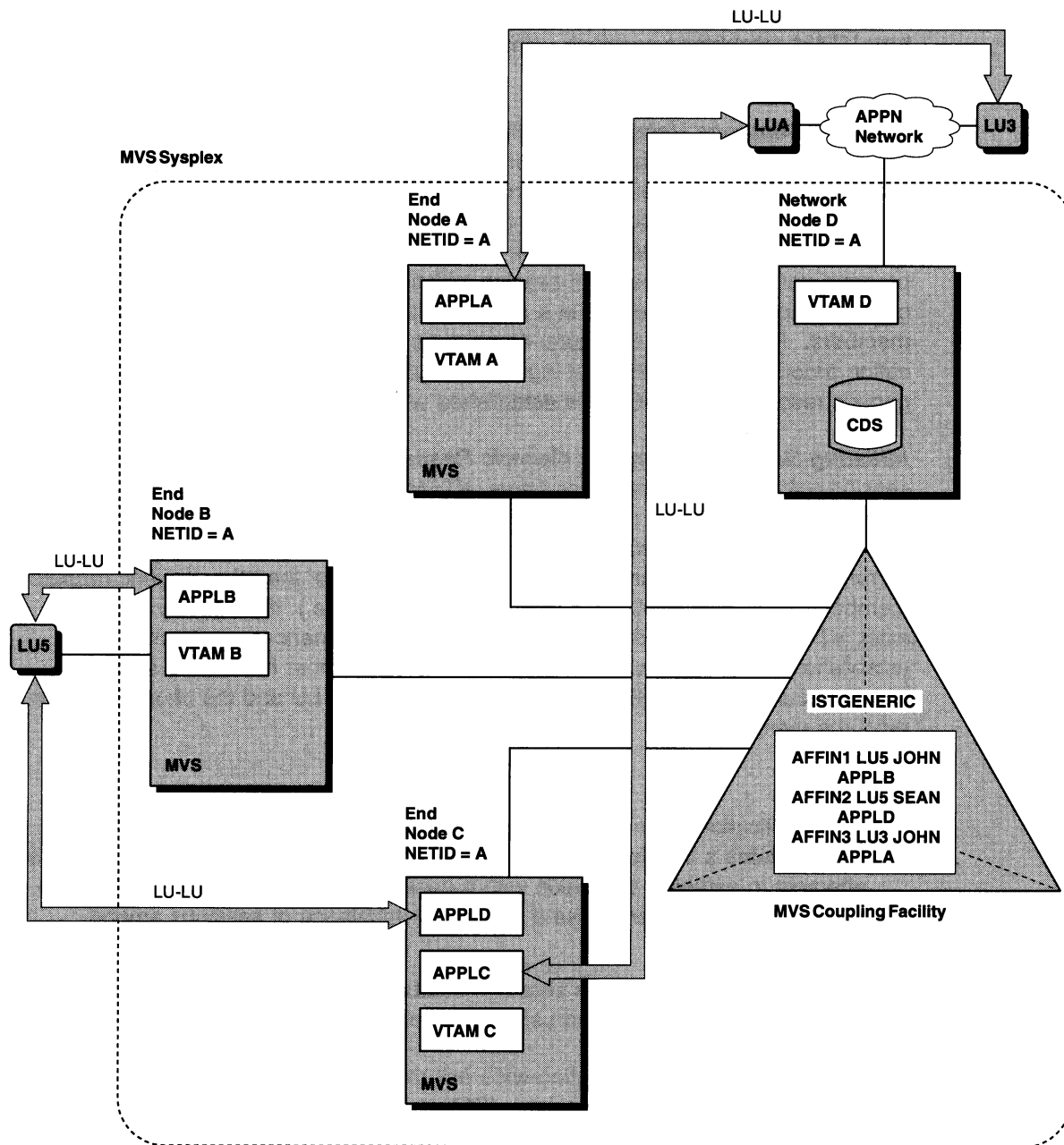


Figure 49. Session Establishment with Generic Resource Members

Initiating Sessions With Generic Resource Members

Using generic resources, VTAM now allows an application program to be known by the generic resource name. The generic resource name is specified on the GNAME operand of the NIB macroinstruction and used when the application program issues SETLOGON OPTCD=GNAMEADD. VTAM uses RACF class VTAMAPPL to validate the generic resource name.

For any particular generic resource name, an LU can initiate a session using the generic resource name or the application program network name of the generic resource member, as shown in the following sections. For more information about how VTAM establishes sessions using the generic resources function, see *VTAM Programming*.

Initiating Sessions Using the Application Program Network Name: If an LU accesses a generic resource member by the application program network name, VTAM establishes the session with that member, regardless of the current workload of that member. In addition, when the first session between an LU and a generic resource member is started using the generic resource member's application program name, any subsequent session initiation attempt from that LU to the generic resource name results in a session with one of the other generic resource members. For example, in Figure 49 on page 187, if LUA initially used the application program name APPLC at logon, any subsequent session from LUA using the generic resource name JOHN is established with APPLA or APPLB.

Initiating Sessions Using the Generic Resource Name: When an LU initiates a session using a generic resource name, VTAM determines whether the LU has an affinity, or mapping, with a generic resource member. If an affinity exists, VTAM establishes the session with the same generic resource member, regardless of current workload. (A session can be started with one of the other generic resource members using their application program network name.) If an affinity does not exist, VTAM determines the session count for each generic resource member, establishes the session with the generic resource member having the smallest session count, and establishes an affinity between the LU and the chosen generic resource member.

Notes:

1. If an application or an LU that is part of a local SNA or local non-SNA major node initiates a session to a generic resource and no affinity exists, VTAM first attempts to establish a session with a generic resource member on the same VTAM node. This can cause a temporary imbalance of sessions among generic resources.
2. The session count includes sessions established using the generic resource name and those established using the application program network name.

VTAM also provides an installation-wide exit routine that can be invoked to override VTAM's selection, if no affinity exists. IBM provides a default exit routine, ISTEXCGR, that can be modified to select the member based on user-specified criteria.

Implementation Considerations

When implementing the generic resources function, the following considerations apply to the generic resources configuration:

- All generic resource members with the same generic resource name must reside within the same generic resource configuration.
- LUs that initiate sessions using a generic resource name may be located outside the generic resource configuration which supports that generic resource name.
- A network node server is required to provide generic resource resolution for resources on end nodes within the generic resource configuration.

- Using two network node servers maintains generic resource resolution capability should one of the network nodes fail.
- All network nodes that are part of the sysplex act as central directory servers (CDSs).
- Performance may be enhanced by placing generic resource members at end nodes in the generic resource configuration. This frees up network nodes to provide routing services.
- An application program can be known by only one generic resource name.
- Generic resource members using the same generic resource name must have the same NETID.
- Generic resource names must be unique within a single network. A generic resource name cannot be identical to:
 - A USERVAR
 - An alias name
 - A real LU name.
- The generic resources function supports multiple copies of the same application program; however, VTAM does not verify that each application program known by a generic resource name provides the same function.
- If your network has a security manager program product installed, the security level for the generic resource name must be the same as the security level for all application programs that are using that generic resource name. The security level of the generic resource name is determined by the first generic resource member that associates itself with that generic resource name. If there is no security product, any application program residing on a VTAM that is connected to an MVS coupling facility can use any generic resource name.
- Using the generic resources function, VTAM distributes sessions across generic resource members rather than evenly across hosts. VTAM evaluates the current usage of each application program, regardless of the owning host. When activating generic resource members, consider the host capabilities. If VTAM A has twice the capability of VTAM B, VTAM A should support twice as many generic resource members.

Connecting to and Allocating Storage for ISTGENERIC

When a VTAM V4R2 node is initialized it may attempt to connect to ISTGENERIC. Before a VTAM node attempts to connect to ISTGENERIC it first checks to see that it is running on a version of MVS which supports the IXLCONN service (a VTAM node connects to ISTGENERIC using the IXLCONN service provided by MVS), and that ISTGENERIC is defined to the active coupling facility resource management (CFRM) policy. The first VTAM node in a generic resource configuration to connect to ISTGENERIC causes the allocation of storage for ISTGENERIC within a coupling facility and also defines the structure attributes associated with ISTGENERIC.

The Structure Attributes Associated with ISTGENERIC

The only attribute defined in the active CFRM policy for ISTGENERIC is a limit on the amount of storage which may be allocated for ISTGENERIC. The set of attributes associated with a given coupling facility structure is dependent upon the type attribute for that structure. The type attribute for ISTGENERIC is defined at the time the first VTAM node connects to ISTGENERIC, and ISTGENERIC is always

created as a coupling facility list structure. MVS provides the IXLLIST service for reading data from and writing data to coupling facility list structures.

To specify the size of ISTGENERIC in the active CFRM policy, use the formula for determining the size of a serialized list structure in the *ES/9000** and *ES/3090* PR/SM* Planning Guide*. The structure attributes used to calculate the size of ISTGENERIC are:

- List structure types Adjunct and Names.
- A list element characteristic of two.
- A list count of four.
- Four lock entries.
- The maximum number of data elements that may be associated with a given list entry in ISTGENERIC is 16.
- VTAM specifies the initial entry-to-element ratio as follows:
 - List entry portion = 50
 - Data object portion = 1

During the rebuild process this ratio may be altered to better fit the actual use of list entries and elements within ISTGENERIC.

- The maximum list-set-entry count. For a formula which can be used to determine the number of list entries needed for a given generic resource configuration, see "Defining a Structure Size for ISTGENERIC" on page 191.
- A lock table entry characteristic of 0.

The structure attributes used to determine the coupling facility preference and exclusion list for ISTGENERIC are:

- Structure rebuilds are allowed for ISTGENERIC.
- The structure disposition for ISTGENERIC is specified as DELETE. This means that ISTGENERIC is deallocated after the last connection to that structure is deleted.
- The connection disposition is specified as KEEP. This means that a connection to ISTGENERIC is not deleted when the corresponding VTAM node disconnects, unless the disconnecting VTAM explicitly indicates that the connection is to be deleted. This means that if a VTAM node fails or is forced to disconnect due to some type of failure, the nodes connection to ISTGENERIC remains intact.
- For ISTGENERIC, list entry control data is specified and tracked in terms of list entries, rather than data elements or list elements.

Once the size of the ISTGENERIC structure has been calculated and the structure attributes associated with ISTGENERIC have been determined, the corresponding CFRM policy can be defined following the directions in *MVS/ESA Setting Up a Sysplex*.

Defining a Structure Size for ISTGENERIC

The only variable parameter for ISTGENERIC is the number of list entries which ISTGENERIC must support for a given generic resource configuration.

The number of list entries which ISTGENERIC must support is equal to the number of generic resource users that the corresponding generic resource configuration is expected to support. A generic resource user is essentially any LU which logs on to any application which is a member of a generic resource. Note, however, that if a single LU establishes sessions with multiple generic resources (for example, LUA logons on to both APPL1 which has the generic name GRAPPL1, and APPL2 which has the generic name GRAPPL2), that LU represents multiple generic resource users (one per generic name involved).

Failure Recovery for the Generic Resource Configuration

There are three types of failures which can cause generic resource data to be lost:

- Failure of a VTAM node; all of the local data of the node is lost.
- Failure of a link between a VTAM node and ISTGENERIC; the VTAM node involved is unable to access or update data in ISTGENERIC.
- Failure of ISTGENERIC; all of the data in ISTGENERIC is lost.

The type of failure recovery provided for each type of failure is described below.

The generic resource function is designed to maintain data integrity when any single type of failure occurs, but does not reliably handle concurrent failures of different types. For instance, if a VTAM node and ISTGENERIC fail concurrently, data is lost. Thus, it is advisable to isolate the coupling facilities from the VTAM nodes (for example, they should not share the same power supply).

In the same way, multiple coupling facilities in the same generic resources configuration should be isolated from each other. If more than one coupling facility is present in a generic resource configuration, a new version of ISTGENERIC can be created if the coupling facility containing the active version of ISTGENERIC fails. Also, if a link failure to ISTGENERIC occurs, ISTGENERIC may be moved to a new coupling facility where it may be accessed by all VTAM nodes in the generic resource configuration.

Failure of a VTAM Node: All local data in a VTAM node that needs to persist through a VTAM failure is replicated in ISTGENERIC. If other VTAMs are present, they provide the necessary cleanup, and generic resources on the failing node can be re-started on another VTAM node. Affinities which should persist through a VTAM failure include LU 6.1 sessions and LU 6.2 sessions with SYNCPT synchronization support. If a failing VTAM has no local persistent data, then a peer VTAM performing cleanup can delete the failed VTAM's connection.

Note: If a VTAM's connection to ISTGENERIC is in failed-persistent state, an operator should not delete it. If the connection is deleted, all data pertaining to the failed VTAM node is deleted.

Also, ISTGENERIC remains allocated until all connections to it are deleted. Since there are no VTAMs to perform cleanup for the last VTAM to disconnect, you do need to delete the last connection to ISTGENERIC to deallocate ISTGENERIC.

Failure of a Link Between a VTAM Node and ISTGENERIC: When a VTAM node loses its link to ISTGENERIC, it initiates a structure rebuild to attempt to rebuild ISTGENERIC on another coupling facility to which all VTAM nodes in the generic resource configuration have access. The rebuild structure process may be stopped by a peer node that is either unable to connect to, or loses its connection to, the new version of ISTGENERIC.

If ISTGENERIC cannot be rebuilt on an accessible node, the VTAM that has no link to the current version of ISTGENERIC disconnects from ISTGENERIC. Its connection to ISTGENERIC is not deleted, and must not be deleted unless that VTAM node has no data in ISTGENERIC that must persist through the failure. Other VTAMs provide the necessary cleanup, and once the failing link is repaired the disconnected VTAM attempts to reconnect to ISTGENERIC.

Failure of ISTGENERIC: When ISTGENERIC fails, each VTAM in a generic resource configuration is notified and attempts to initiate a structure rebuild to create a new version of ISTGENERIC. The new version of ISTGENERIC is replenished from the local data of each VTAM node in the generic resource configuration.

Structure Rebuild

The structure rebuild process is used to create a new copy of a coupling facility structure, either in the same coupling facility or in a different coupling facility, and to store in this new copy of the structure all of the data from the old copy of the structure. During the structure rebuild process for ISTGENERIC, data is either copied by a VTAM node in the generic resource configuration from the old version of ISTGENERIC into the new version of ISTGENERIC, or all VTAM nodes in the generic resource configuration each supply the new version of ISTGENERIC with local data.

Initiating a Structure Rebuild: There are several reasons why a structure rebuild may be initiated.

- An MVS operator can initiate a structure rebuild for ISTGENERIC.
- A VTAM node in a generic resource configuration may initiate a structure rebuild if it loses access to ISTGENERIC.
- A VTAM node in a generic resource configuration may initiate a structure rebuild when notified that ISTGENERIC has failed.

Stopping a Structure Rebuild: Once a structure rebuild has been initiated, it can be stopped for any one of several reasons.

- An MVS operator can stop a structure rebuild for ISTGENERIC.
- MVS automatically stops a rebuild if the new structure being created fails or becomes inaccessible.
- A VTAM node in a generic resource configuration may stop a structure rebuild if it loses access to the new version of ISTGENERIC. A VTAM node never stops a structure rebuild once the old version of ISTGENERIC has failed.
- A VTAM node in a generic resource configuration may stop a structure rebuild when it determines that all of the data in the old version of ISTGENERIC may not be available for creating the new version of ISTGENERIC.

Once a structure rebuild is stopped, the old version of ISTGENERIC continues to be used (if possible) and the new version of ISTGENERIC, if allocated, will be deallocated.

Routine Maintenance for VTAM Nodes

VTAM maintains affinities between LUs and their generic session partners both at the node of the generic session partner and in the coupling facility. Some such affinities may need to persist across VTAM node failures. Before bringing a VTAM down for routine maintenance, remove the need for any affinities and the affinities themselves. This lessens the effects of a possible failure of ISTGENERIC when a VTAM node is being serviced.

LU 6.2 in an XRF Environment (MVS)

USERVAR's process of using a generic name to represent a service does not work in an active LU 6.2 environment because LU 6.2 cannot tolerate LU name substitutions. However, support for LU 6.2 in an XRF environment allows generic name substitution, but only in a recovery environment.

The LU 6.2 application uses a USERVAR ID during recovery to transfer its users from one application program to a backup application program. This provides faster backup for failing applications.

To use this support, you need to identify the USERVAR to be used for LU 6.2 communications.

Be aware of the following restrictions:

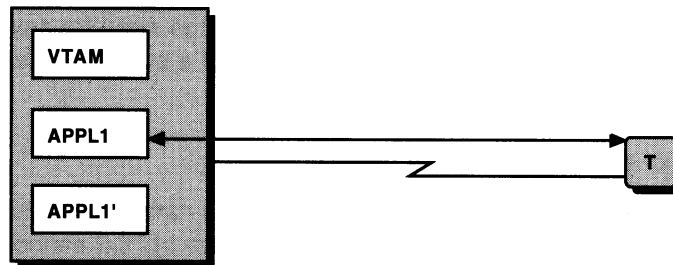
- You can assign only one USERVAR ID to an LU 6.2 application program.
- You can have only one LU 6.2 application program as the primary application program to provide a service.
- The operator can modify the USERVAR ID for LU 6.2, but the command must be issued from the host system of the application program providing the service.
- If an incorrect USERVAR ID is assigned to an active LU 6.2 application program, the ACB must be closed and then reopened with the correct USERVAR ID.
- Even sessions that request a bind using the real LU name must still carry the USERVAR ID.

Persistent LU-LU Sessions (MVS, VSE)

If an application program fails or is brought down, VTAM can retain active sessions, allowing the same or another application program to reconnect to the sessions, avoiding the need to re-establish the sessions. If an application program fails, it can reconnect to the retained sessions when it recovers. Also, another application program can take over the sessions.

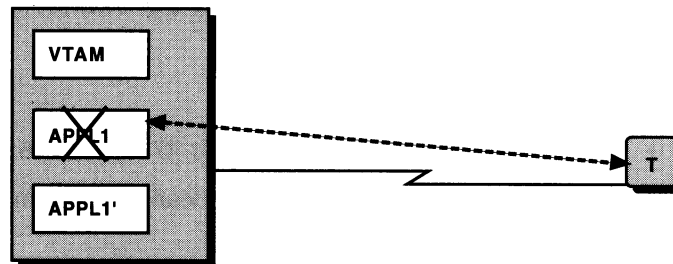
For an application to be capable of persistence, code PERSIST=YES on the ACB macroinstruction. To enable persistence, the application program must specify OPTCD=PERSIST on the SETLOGON macroinstruction. The application program can also disable persistence by issuing SETLOGON OPTCD=NPERSIST at any time.

When a persistence-enabled application program fails, VTAM retains the sessions, saves the allocated resources and control blocks, and shields the network from knowledge of the application program failure. VTAM stores the incoming data so that the network views the session as active but not currently responding. When the failed application program restarts or another application program takes over, VTAM reconnects the sessions. The backup of a persistence-enabled application program is shown in Figure 50.



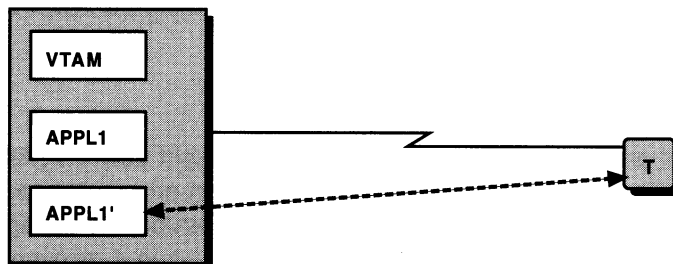
1. APPL1 has initiated a session with terminal T. The session is persistent capable and enabled. APPL1 specified persistence on both the OPEN ACB (PERSIST = YES) and the SETLOGON (SETLOGON OPTCD = PERSIST) macroinstructions.

Figure 50 (Part 1 of 3). Application Program Backup Using Persistent LU-LU Sessions



2. APPL1 fails. The session is pending recovery. If no timer is specified (PSTIMER), the session can remain pending recovery indefinitely. However, the retained session can be terminated by a VARY NET INACT command.

Figure 50 (Part 2 of 3). Application Program Backup Using Persistent LU-LU Sessions



3. The operator or program operator detects that APPL1 has failed and uses APPL1' as a backup. APPL1' issues an OPEN macroinstruction with the same ACB name APPL1 issued. It then restores the session. Takeover is transparent to terminal T.

Figure 50 (Part 3 of 3). Application Program Backup Using Persistent LU-LU Sessions

For more information on enabling an application program to support persistent LU-LU sessions, see “Restoring Modes and Any Associated Persistent LU-LU Sessions (MVS, VSE)” in *VTAM Programming for LU 6.2* and “Using Persistent LU-LU Session Support (MVS, VSE)” in *VTAM Programming*.

VSE With persistent sessions, data space sizes can expand. To limit the size to which a data space can expand, use the DSPACE operand on the EXEC JCL statement for a VTAM application program. This prevents a data space from expanding to the point that it uses all available virtual storage. For more information on limiting data space expansion, see “Defining Data Space Virtual Storage” on page 115.

VTAM Common Network Services (MVS, VM)

When you use VTAM Common Network Services (VCNS) with the ES/9000 series communication adapter to send data across an X.25 packet switched data network, specify the VCNS=YES operand on the APPL definition statement. This allows sharing the physical connectivity of the communication adapter between SNA and VCNS data traffic.

You can use VTAM Common Network Services (VCNS) with the IBM 3172 interconnect controller to send data to a local area network. Specify VCNS=YES on the APPL definition statement to allow sharing the physical connectivity of the communication adapter between SNA and VCNS data traffic.

For more information on VCNS, see “VTAM Common Network Services (VCNS)” on page 246.

Cross-Memory Application Program Interface (API) (MVS)

Sessions can be established by an application program running in an address space other than that which opened the application program’s ACB. This enables the application program to be represented to VTAM by one ACB (that is, APPL definition statement) but span multiple address spaces. The application program can also issue VTAM requests to send and receive data while running in a different address space than the one in which the session was originally established.

For information on enabling an application program to support cross-memory API, see “Cross-Memory Application Program Interface (API) Support (MVS)” in *VTAM Programming*.

Allocating Private Storage

VM The MAXPVT operand on the APPL definition statement enables you to tell VTAM how much it can use of the private storage allocated to the application program’s task. VTAM uses this storage for queueing data that has come into the host for a session and for which no RECEIVE has been issued. In most cases, each task is associated with only one application program. If a task involves more than one application program, the effective MAXPVT value for the task is the sum of the values specified for all the application programs opened under the task.

MVS, VSE All incoming data is queued in a VTAM-owned ESA data space, not in private storage. The MAXPVT operand therefore has no effect on data queued for an application.

Communicating with Start-Stop Devices

When communicating with a start-stop device that is supported by the Network Terminal Option (NTO) or the X.25 NCP Packet Switching Interface (NPSI), special data stream characteristics of the non-SNA terminal might need to be specified for use by an application program. You indicate these characteristics by specifying the appropriate value on the TERM operand of the PU or LU definition statement that defines the virtual physical or logical unit. The application program can get information about these characteristics by using the INQUIRE OPTCD=DEVCHAR macroinstruction. It can also get it from the CINIT RU, which is queued when a logon is received from one of these devices.

LU 6.2 Application Programs

LU 6.2 enables a VTAM application program to communicate with non-host application programs on an advanced program-to-program communication (APPC) level. Without LU 6.2, the non-host program usually must emulate a terminal when communicating with a VTAM application program, using protocols that are not well-suited for application programs and that diminish the non-host application program's ability to handle errors and to initiate and end communications.

LU 6.2 protocols are designed to optimize use of sessions and improve the efficiency of data transfer through the network. LU 6.2 can also simplify the program logic needed to establish communications with other application programs and to send and receive data. Though VTAM's support simplifies the task of writing application programs that use LU 6.2, be aware that enabling VTAM support requires some LU 6.2 functions to be implemented by an application program.

You can write or modify host application programs to use LU 6.2 protocols throughout the network.

Enabling LU 6.2 Support

You enable LU 6.2 support for an application program by specifying the APPC=YES operand on the APPL definition statement. The application program must use the APPCCMD macroinstruction to send and receive data on LU 6.2 sessions and must allow VTAM to manage LU 6.2 session establishment and termination. This means that an application program cannot use non-LU 6.2 VTAM macroinstructions such as OPNDST, SEND, or RECEIVE on LU 6.2 sessions; however, the same application program can use non-LU 6.2 macroinstructions to communicate on non-LU 6.2 sessions. For information on macroinstructions for LU 6.2 sessions, see *VTAM Programming for LU 6.2*.

Existing application programs that have implemented LU 6.2 without VTAM's APPCCMD support (and consequently use the restricted non-LU 6.2 macroinstructions) should not have APPC=YES on their APPL definition statement. They do not work if it is included. To take advantage of VTAM support, they must be migrated to use the APPCCMD macroinstruction.

LU 6.2 Sessions

Following is information about how LU 6.2 sessions operate.

Session Establishment and Termination: VTAM handles the tasks of session establishment and termination for LU 6.2 sessions for those application programs that use LU 6.2 support. Application programs explicitly issue conversation establishment and termination requests. VTAM automatically sets up and takes down

sessions in response to these requests, according to the session limits the application programs have requested and negotiated. The application program's main involvement with session management is in establishing limits on the number of sessions that it can have with another LU in the network.

Conversations: Sessions between LU 6.2 session partners are used by transaction programs that operate in each partner LU to perform data exchange. Rather than sessions, the basic communication link between logical units using LU 6.2 protocols is a conversation. A conversation is the exclusive use of a session between two partner LUs for some unit of work. Sessions between LU 6.2 session partners are serially reusable. When one conversation ends, another conversation can use that same session between the partner LUs.

Sending and Receiving Data: If a session is full-duplex, conversations on that session can be full-duplex or half-duplex. If a session is half-duplex, conversations on that session must be half-duplex. In full-duplex conversations, an LU can send and receive data simultaneously. In half-duplex conversations, only one LU can send data at a given time. One transaction program in the conversation must be designated as the sender and the other as the receiver. Sessions between logical unit partners are designated as contention-winner or contention-loser sessions. If the two LU partners attempt to use a session simultaneously, the designated contention winner for the session has priority for the use of the session. An application program can allocate a conversation over a session for which it is designated the contention loser, but the logical unit partner must give the application program permission.

Session Characteristics: Session characteristics are determined by the mode name associated with the session. Each session between LU 6.2 partners is related to a specific mode name. The mode name represents a set of characteristics associated with that session. All other sessions between the LU 6.2 partners that use that mode name are assumed to have the same set of session characteristics.

LU 6.2 Session Limits

VTAM maintains a set of session limits for partner logical units for each mode name used. These session limit values are used if a partner LU attempts to negotiate a change in the current number of sessions. If VTAM receives a request from the other partner LU to change the number of sessions for its LU 6.2 logical unit, VTAM uses the session limit values to determine whether to accept or reject the requested change. Note that the partner LU may be unable to accept increases in the session limits. Therefore, the end result of any change requests can still mean that the session limits do not increase.

If a session is available for use by a conversation, the process of starting and ending a conversation requires fewer system resources than starting and ending a session. Ideally, therefore, application programs should set up their session limits so that VTAM can establish a number of sessions with other LUs that will be relatively long-lived. Applications can then treat these sessions as resources that support conversations.

Requesting Session Limit Changes: Application: An LU 6.2 logical unit requests that its partner LU alter the number of sessions for a given mode name by sending a change-number-of-sessions (CNOS) request. A VTAM application program can request that its partner LU change the number of sessions. The appli-

cation program does this by issuing an APPCCMD macroinstruction with the CONTROL=OPRCNTL and QUALIFY=CNOS operands specified.

Requesting Session Limit Changes: Operator: You can enable the operator to modify CNOS parameters by specifying OPERCNOS=ALLOW on the APPL definition statement. The parameters can then be changed by a MODIFY CNOS command, which functions the same as an APPCCMD macroinstruction with the CONTROL=OPRCNTL and QUALIFY=CNOS operands specified. The operator can then also use the MODIFY DEFINE command, which functions the same as an APPCCMD macroinstruction with the CONTROL=OPRCNTL and QUALIFY=DEFINE operands specified. The application is notified when the operator-issued commands have completed.

Coding Session Limits: These session limit values can be set for a VTAM application program by coding specific operands on the APPL definition statement. The VTAM application program can also establish these session limit values and override the specifications on the APPL definition statement by issuing an APPCCMD macroinstruction with the QUALIFY=DEFINE operand specified. Following are the unique session limit values that are used:

- The overall maximum number of sessions between two partner LUs for a given mode name, which you specify on the DSESLIM operand of the APPL definition statement
- The minimum number of sessions for a given mode name for which the application program is the contention winner, which you specify on the DMINWNL operand of the APPL definition statement
- The minimum number of sessions for a given mode name for which the application program might be the contention loser (the partner LU is the contention winner), which you specify on the DMINWNR operand of the APPL definition statement.

The sum of the DMINWNL and DMINWNR values must always be less than or equal to the maximum number of sessions specified on the DSESLIM operand. The AUTOSSES operand on the APPL definition statement can be used to have VTAM automatically activate a specific number of contention-winner sessions for a given mode name for the application program. VTAM establishes these sessions when the first CNOS request is issued by the application program for a partner LU using a given mode name.

Session Limits and Deactivation: Using CNOS, an LU 6.2 session partner can request that the session limits be set to zero. Then, the sessions that are active and not used for a conversation are deactivated. The LU partner that is responsible for deactivating the sessions for a given mode name is determined by an indicator in the CNOS request. The DRESPL operand on the APPL definition statement can be used to indicate if the VTAM application program accepts the responsibility for deactivating the sessions (ALLOW) or if VTAM assigns that responsibility to the partner LU when the CNOS request is sent (NALLOW).

When sessions between partner LUs are to be deactivated, the existing conversations are not disrupted, but subsequent allocation requests for conversations can be queued. Prior to deactivating the sessions for a given mode name, the partner LUs can indicate whether conversations that are queued for a session are to be processed or discarded. The DDRAINL operand on the APPL definition statement indicates whether the VTAM application drains the queued allocation requests when

deactivating sessions. If you code DDRAINL=ALLOW, the conversations that are queued for a session for the given mode name are processed before deactivating the sessions. If you allow conversations to be drained, new conversations are allocated a session if the allocation request occurs prior to the deactivation of all of the sessions for the given mode name. If you code DDRAINL=NALLOW on the APPL definition statement, VTAM does not allow the application program to drain queued session requests and the conversation requests are rejected.

Managing LU 6.2 Sessions with Operator Commands

Session characteristics, such as session limits, LU capability, and LU responsibilities, must be managed for every conversation involving an LU 6.2 application program. These characteristics can be controlled by the application program and through operator commands.

Operator commands manage conversation resources in the following ways:

- Establishing or modifying new session limits between an application and partner LU (MODIFY CNOS)
- Establishing or modifying session limit values VTAM uses to negotiate CNOS requests initiated by a partner LU for a specific application (MODIFY DEFINE). These values are stored in the logon mode table.

You can establish session limits using the MODIFY CNOS operator command. After session limits are established, actual sessions can be started.

Session limits must also be established for a mode name before conversations that use that mode can be allocated. When a request for a session is made, the requested session limits can be accepted by the partner LU or the session limits can be changed to new values. The process for accepting or changing proposed session limits is called negotiation. Logon mode table values are used in session limits negotiation. Negotiation is handled for the application program by the MODIFY DEFINE operator command. For more information on session negotiation, see “Negotiating Session Limits” in *VTAM Programming for LU 6.2*.

LU 6.2 Security

VTAM's LU 6.2 support provides the following security functions for your network:

- **MVS** Encryption facility
- LU 6.2 conversation-level verification (user ID verification)
- LU 6.2 session-level LU-LU verification.

MVS The encryption facility protects data passing over lines between network resources by permitting enciphering and deciphering of data for LU-LU sessions. For an application program or peripheral node logical unit to have cryptographic sessions, the host processor must support cryptography. For more information on the encryption facility see “Encryption Facility ” on page 168.

LU 6.2 user ID verification is a conversation-level security protocol, taking place at the time a conversation is started. For more information on LU 6.2 conversation-level security, see “Conversation-Level Security” in *VTAM Programming for LU 6.2*.

LU 6.2 session-level LU-LU verification is a session-level security protocol, that is used to verify the identity of each logical unit at the time the session is activated. LU 6.2 session-level LU-LU verification provides the ability to verify the identity of

an application program's partner LUs during the activation of sessions between type 6.2 LUs. VTAM-generated random data is encrypted using one of the data encryption standard algorithms. The encryption key is a password associated with each LU-LU pair. The encrypted data is carried on both the session activation request and the response so that each LU partner can verify the other partner for the session. The passwords used for session-level LU-LU verification are not coded on any VTAM definition statements but are implemented through an external security management product, such as RACF. If you plan to use LU 6.2 session-level LU-LU verification, a security management product (such as RACF 1.9 or later) must be installed and active. In addition, a profile for the LU needs to be in the security management's database. With RACF 1.9, the APPCLU class needs to be active and a profile of the LU needs to be in the APPCLU class. For an example of the appropriate RACF coding, see the *RACF Security Administrator's Guide*.

During activation of LU 6.2 sessions involving control points, the VERIFYCP start option specifies whether VTAM performs session-level LU-LU verification.

The VERIFY and SECLVL operands on the APPL definition statement identify the level of partner-LU security verification.

If the application program is the PLU and an LU-LU password is defined for the partner LU, VTAM requests that LU-LU verification be performed during session activation. If a password is not defined, LU-LU verification is not requested.

If the application program is the secondary logical unit, one of the following occurs:

- If the session activation request specifies LU-LU verification and the LU-LU password is defined, verification is performed.
- If the session activation request does not specify LU-LU verification and no LU-LU password is defined, session activation continues without verification.
- If neither of the previous cases applies, VTAM rejects the session activation request.

Note: Specifying VERIFY=OPTIONAL does not restrict the ability of a logical unit without a corresponding LU-LU profile to establish sessions with this application program.

If using RACF as your external security management product, the MODIFY PROFILES command enables you to reload an active application program's set of existing defined RACF profiles. However, you cannot change the RACF profile with MODIFY PROFILES, only refresh it. The RACF profile contains the LU-LU password and only someone with RACF security clearance can change it. This can be helpful when the password for an LU-LU pair has been changed or when session activation errors are occurring. The profile changes affect only those sessions that are started after using the command; active sessions are not affected.

LU 6.2 Conversation-Level Verification

LU 6.2 provides various levels of conversation-level security for user ID verification. The SECACPT operand on the APPL definition statement specifies the degree of security that an application program accepts on requests that are received from another partner LU. For coding the SECACPT operand, see "SECACPT" in the *VTAM Resource Definition Reference*.

LU 6.2 Sync Point Services

LU 6.2 sync point services is a session-level protocol. The maximum service level parameters are exchanged at session activation. However, when the conversation is allocated, the service level can be lower than the parameters exchanged at session activation. This can be caused by the inability of the other logical unit to support a higher service level. Synchronization of protected resources is the application program's responsibility. It is a means of recovery from errors or failures that allows the data network to divide its transmissions into logical units of work and commit or back out the information in these divisions of work.

The sync point function is divided between VTAM and a VTAM application program. LU 6.2 provides the facilities to allow the application program to implement a sync point facility by supporting the flow of information between application programs and the ability to control a session by rejecting, holding, or releasing a session based on a session identifier.

MVS, VSE Persistent LU-LU session support does not allow LU 6.2 sync point sessions to be retained. Instead, they end when an application fails and they cannot be recovered.

The SYNCLVL operand on the APPL definition statement indicates the level of synchronization for this application program. If you do not code SYNCLVL, confirmation requests and sync point processing are not supported.

On the APPL definition statement, you can also specify (in conjunction with sync point support) when the ATTN exit for an LU 6.2 application program is to be scheduled. This exit routine can be invoked when the last session (ATNLOSS=LAST) is deactivated between an LU and MODE (LU-mode) pair. The ATTN exit can also be invoked for all session deactivations by specifying a value of ALL on the ATNLOSS operand. The LU 6.2 application program can use this operand without implementing sync point support.

For more details on sync point support, see "Synchronization Point Services" in *VTAM Programming for LU 6.2*.

Selective Termination of Idle LU 6.2 Sessions

You can limit the use of some network connections, such as lines. Defining a limited network connection enables you to limit the use of the connection and keep only active sessions on that connection. Sessions that are not active are terminated. If all sessions are terminated, then the connection is terminated. The best network connections to define as limited are lines and PUs whose cost is determined by the length of time a connection exists. Defining these as limited can help reduce switched line connect charges.

When a session traverses a limited connection, the session can be terminated if no conversation is detected for a set period of time. You must specify the time-out on the LIMQSINT operand of the APPL definition statement. LIMQSINT specifies how long (in seconds) unused limited resource sessions will remain on the queue before they are terminated.

Note: Only LU 6.2 conversations are affected by limited resource definition. Non-LU 6.2 sessions are unaffected and cannot be limited.

To use LU 6.2 limited resource management:

1. Determine which network connections you want to define as limited.
2. Define the network connections to VTAM by coding LIMRES=YES on the GROUP, LINE, or PU statement. LIMRES is valid for PU types 1, 2, and 2.1 only.
3. Determine the time interval for VTAM to search the queue and define the interval to VTAM using the LIMQSINT operand on the APPL definition statement.

You can specify a line, a group of lines, or a PU as a limited resource for the following major nodes:

- NCP
- Channel-attachment (SDLC switched and non-switched)
- **VM,VSE** Packet (SDLC switched and non-switched)
- **VM,VSE** LAN
- External communication adapter
- Switched (PU only)
- MODEL (PU only).

After you define the resource to be limited, set the time interval for the queue search. As a starting point, you should specify a time interval at least 1 second less than half of the shortest line time cost interval. For example, if the line time cost interval is 120 seconds, you should specify 59 seconds on the LIMQSINT operand. You might need to adjust the time interval for different applications.

Attaching NCP to VTAM

This section contains the following subsections:

- NCP Definition Statements for VTAM
- NCP Definition Statements for NCP
- Channel-Attached NCP
- Link-Attached NCP.

NCP Definition Statements for VTAM

The PCCU, BUILD, and HOST definition statements in the NCP major node can contain information for VTAM.

PCCU Definition Statement

The programmed communication control unit (PCCU) definition statement identifies the communication controller where the NCP is loaded. It is required in VTAM systems and defines the VTAM functions that are provided for this NCP. Code at least one PCCU definition statement for each NCP.

Sample PCCU Definition Statement: Following is a sample PCCU definition statement:

```

PCCU116  PCCU  AUTODMP=NO,          AUTOMATIC DUMP OPTION
              AUTOIPL=NO,          AUTOMATIC RE-IPL OPTION
              AUTOSYN=YES,         SYNCHRONIZE WITH ACTIVE NCP
              CUADDR=016,          3745 IPL ADDRESS
              LOADFROM=EXTERNAL,    LOAD 3745 FROM 3745 DISK
              DUMPLD=YES,          DUMpload USING 3745/3720 DISK
              DUMPDS=NCPDUMP,       NCP DUMP DATA SET DDNAME
              MDUMPDS=MOSSDUMP,     MOSS DUMP DATA SET DDNAME
              CDUMPDS=CSPDUMP,      CSP DUMP DATA SET DDNAME
              MAXDATA=5000,         MAXIMUM PIU SIZE
              NETID=USIBMCS,        NETWORK IDENTIFIER
              VFYLM=YES,            NOTIFY OPERATOR IF WRONG LOAD MODULE
              VFYC=YES,             VERIFY LOAD MODULE AND RRT MATCH
:
              SUBAREA=116          OWNING SSCP SUBAREA

```

SUBAREA Operand: The SUBAREA operand associates the communication controller with a VTAM host when the NCP is activated. If each host provides the same function for the NCP, only one PCCU definition statement is necessary, but the operator might need to override unique attachment requirements.

Multiple PCCU Definition Statements: If you code multiple PCCU definition statements, VTAM compares its network identifier and subarea number to the values of the NETID and SUBAREA operands on all the PCCU definition statements to locate the one that applies to it. However, VTAM selects this PCCU definition statement based on the following rules:

1. The VTAM host processes the first PCCU definition statement whose NETID and SUBAREA operands match VTAM's.
2. If no such PCCU definition statement exists, the VTAM host processes the first one that has the same NETID as VTAM and that does not have a SUBAREA.
3. If a PCCU definition statement as described in step 1 or 2 does not exist, the VTAM host processes the first PCCU definition statement without a NETID operand but with a SUBAREA value equal to VTAM's.
4. Finally, if a PCCU definition statement as described in step 1, 2, or 3 does not exist, the VTAM host processes the first PCCU statement for which neither the NETID nor SUBAREA operand is coded.

The NCP for an IBM 3745 or 3720 Communication Controller can be loaded from either of the following:

- The NCP load library at the host (LOADFROM=HOST)
- The 3745 or 3720 disk (LOADFROM=EXTERNAL).

When the VARY ACT command is issued to activate NCP, the LOADFROM operand on the PCCU definition statement is used if the LOADFROM operand is not specified on the VARY ACT command.

Automatic Dump and Reload: The AUTODMP and AUTOIPL operands on the PCCU definition statement inform VTAM whether to automatically dump and reload the NCP if an error is encountered. To have an IBM 3745 or 3720 Communication Controller dump and reload the NCP from its disk, do one of the following:

- Code the DUMPLD=YES operand on the PCCU definition statement.

- Specify the DUMPLOAD=YES operand on the VARY ACT command.

These operands are then ignored if the operation is successful. The NCP dump can then be transmitted to the VTAM host from the communication controller disk at a later time.

If the communication controller is a 3725 or the dump is not stored on the communication controller disk, it is transmitted to the VTAM host and stored in the file identified by the DUMPDS operand on the PCCU definition statement. If you specify the CDUMPDS and MDUMPDS operands on the PCCU definition statement, they identify the files in which a CSP and a MOSS dump, respectively, are to be stored in the VTAM host. If you do not specify these operands, the corresponding dump is stored in the file identified by the DUMPDS operand.

Verifying the Load Module: When you load the NCP, you can use the AUTOSYN operand of the PCCU definition statement to request VTAM to compare the name of the load module currently in the communication controller to the NEWNAME operand on the BUILD definition statement. If you code AUTOSYN=YES, VTAM uses the current NCP load module; and if you code AUTOSYN=NO, the operator must determine whether the communication controller should be reloaded with a new copy of the NCP. If the NCP load module and the NEWNAME operand do not match, the VFYLM operand determines whether the operator is to be informed of the mismatch (VFYLM=YES) or whether the operator must simply reload the communication controller with the different load module (VFYLM=NO).

For a channel-attached NCP, the CUADDR operand of the PCCU definition statement indicates the channel unit address of the communication controller that VTAM should use to load the NCP.

Verifying Time Stamps: When the NCP/EP definition facility (NDF) is used to generate NCP, a resource resolution table (RRT) is created in addition to the NCP load module. At initialization, VTAM reads the RRT to determine what resources are contained in the NCP load module. VTAM examines the time stamp of the RRT and verifies that it corresponds to the NCP load module associated with the RRT.

The VFYC operand on the PCCU definition statement enables you to specify what action VTAM should take if the time stamps do not match. If you code VFYC=NO or allow it to default and VTAM detects a correlator mismatch, VTAM sends a message to the operator explaining that the NCP is being loaded due to the mismatch. If you specify VFYC=IGNORE, VTAM continues the NCP activation process regardless of the time stamps of the NCP load module and the RRT; no message is issued.

BUILD Definition Statement

The operands on the BUILD definition statement are primarily used by the NCP generation process. The SUBAREA operand identifies the subarea address of the NCP to VTAM. The subarea address number specified must be within a subarea address range that is acceptable to VTAM; this depends on whether the network contains non-extended network addressing nodes, extended network addressing nodes, or extended subarea addressing nodes.

Sample BUILD Definition Statement: Following is a sample BUILD definition statement:

```

BUILD      BUILD BFRS=240,          SIZE OF NCP BUFFERS
           MEMSIZE=4M,             3745 MEMORY SIZE
           MODEL=3745-210,         CONTROLLER IS AN IBM 3745
           VERSION=V5R2.1,        NCP VERSION
           USGTIER=4,             USAGE TIER 4

:

           LOCALTO=1.5,           (NTRI) LOCAL ACKNOWLEDGEMENT TIMER
           MXRLINE=2,             (NTRI) 2 TICS ATTACHED
           MXVLINE=16,           (NTRI) 16 PUS MAXIMUM
           REMOTTO=1.5,          (NTRI) REMOTE ACKNOWLEDGEMENT TIMER

:

           NETID=CSSNET,          NETWORK ID
           SUBAREA=111,           NCP SUBAREA NUMBER
           SALIMIT=255,           MAXIMUM SUBAREA ADDRESS LIMIT
           ERLIMIT=8,             USE ONLY 8 EXPLICIT ROUTES
           NEWNAME=N111A2,        NCP LOAD MODULE NAME-NCP 111, MONTH 10, CYCLE 2
           PUNAME=NCP111,         NCP PU NAME
           LOADLIB=NCPLoad        LOAD MODS IN NCPLoad LIB

```

Subarea Addressing: In the preceding sample, the MAXSUBA operand is not coded, which means that the network does not contain non-extended network addressing nodes. The SALIMIT operand, which is used in an extended subarea addressing node, limits the subarea address range for the sample NCP to 255 subareas. This subarea address range is the same as for an extended network addressing environment. The maximum subarea address range for an attached VTAM can be greater than 255; however, because SALIMIT=255 for the NCP, its subarea address must be less than or equal to that limit.

You can limit the number of explicit routes that can be used in an extended subarea network by coding the ERLIMIT operand on the NCP BUILD definition statement. Although VTAM supports up to 16 explicit routes, if the number of explicit routes in the NCP is limited to 8, VTAM must use explicit route numbers 0 through 7 to communicate with that NCP.

For a single-domain network, the NETID operand on the BUILD definition statement must be the same as the network identifier used to start VTAM. For information on coding the BUILD definition statement, see the *NCP, SSP, and EP Resource Definition Reference*.

Identifying the NCP Load Module: To identify the NCP load module:

1. Code the LOADLIB operand on the BUILD definition statement to identify the file in which the NCP load module is stored.
2. Code the NEWNAME operand on the BUILD definition statement to identify the name of the load module. This is also the name used to file the NCP in VTAMLST.
3. If multiple copies of an NCP must be maintained at a single host, alter the load module name by changing the value of the NEWNAME operand.
4. You can refer to the NCP using the same physical unit name by coding the same value on the PUNAME operand for each different NCP load module.

5. Activate the NCP by specifying `ID=puname` and `LOADMOD=newname`. If you specify `ID=puname` on the `VARY ACT` command, a `DISPLAY` command for the NCP is issued specifying `ID=puname`.

If the NCP is to be automatically activated using a VTAM configuration list (CONFIG start option), the NCP major node name must be the same as the load module name specified on the `NEWNAME` because it is filed in VTAMLST using that name. For example, in the preceding BUILD definition statement, you can always refer to the NCP by its physical unit name (PUNAME) of NCP111. The load module name specified on the `NEWNAME` operand can contain other information such as the month (October is 10 or X'A') and the cycle number of the NCP generation for that month.

You can also maintain different load modules for the same NCP by filing the NCPs under different temporary names. Before activating one of them as a VTAM major node, rename the member to match the load module name (`NEWNAME`) on the BUILD definition statement.

HOST Definition Statement

VTAM examines the HOST definition statement to determine the amount of data that VTAM must be prepared to receive from the NCP over the channel. The value VTAM uses is the `MAXBFRU` specification times the size of VTAM buffers (IOBUF size). In exchange identification (XID) processing, VTAM informs NCP of the maximum amount of data that it can receive from the NCP. NCP also informs VTAM of the maximum amount of data that it can receive. The value NCP uses in the XID process is the NCP buffer size (BFRS) times the value specified on the `TRANSFR` operand of either the BUILD or LINE definition statements. VTAM compares this value to the `MAXDATA` operand on the PCCU definition statement and returns the smaller of the two values.

Sample HOST Definition Statement: Following is a sample HOST definition statement:

```
VTAM116  HOST  BFRPAD=0,          BUFFER PAD FOR VTAM
               MAXBFRU=50,        INITIAL UNITS FOR INPUT
               SUBAREA=116,        HOST SUBAREA
               UNITSZ=512          BUFFER UNIT SIZE
```

Note: With NCP V4R2 and later releases, the `UNITSZ` specification for an NCP is ignored by VTAM and is no longer used in channel programming. If specified, it should match the size specified for the IOBUF buffer pool start option, but it does not affect the success or failure of link establishment.

Specifying a Value for MAXDATA: The `MAXDATA` operand should be smaller than or equal to the size of the largest PIU the network will handle. Use the following criteria for choosing the value of `MAXDATA`:

- `MAXDATA` is related to the `BFRS` and `TRANSFR` operands of the BUILD definition statement. `MAXDATA` should not exceed the product of `BFRS` and `TRANSFR`, minus 18 ($BFRS \times TRANSFR - 18$).
- `MAXDATA` should not exceed the size of VTAM's IOBUF buffer pool.
- `MAXDATA` should not exceed the product of the `MAXBFRU` and `UNITSZ` values for the NCP.

- If a MODIFY LOAD command will be issued for a local or remote NCP, the size of MAXDATA must be at least 2082, which is 2048 plus the size of the PIU header (34).
- Although MAXDATA applies only to the PCCU statement for a channel-attached NCP, its value should not exceed the capacities of any link-attached or cross-domain NCPs. An SNA path error can occur if MAXDATA exceeds the capacity of any NCP along the route taken by a message.

For more information on selecting the values for VTAM, refer to Chapter 11, “Tuning VTAM for Your Environment” on page 565.

NCP Definition Statements for NCP

The following definition statements contain information that the NCP needs.

BUILD Contains information about NCP's characteristics (such as controller storage size, subarea address of the NCP, and characteristics of channel adapters).

SYSCNTRL

Identifies the dynamic control facilities included in the NCP. These facilities allow the NCP to execute requests from VTAM to change certain NCP parameters or to determine the status of resources such as lines and stations.

HOST Contains characteristics of VTAM. Knowledge of these characteristics is needed to conduct channel operations for communication with VTAM.

LINE, PU, and LU definition statements

Define SDLC resources to NCP. The LINE, CLUSTER, and TERMINAL definition statements define BSC resources to NCP. The PUDRPOOL and LUDRPOOL definition statements define pools of addresses. These addresses are used to support NCP dynamic reconfiguration (both PUDRPOOL and LUDRPOOL) and switched connections to NCP (LUDRPOOL only).

Channel-Attached NCP

MVS You can either statically or dynamically define a channel-attached NCP. See “Dynamic Configuration of Channel-Attached Devices ” on page 147 for information on implementing dynamic definitions.

To define your channel-attached NCP as a type 2.1 device, and define the logical unit either dynamically or statically, see “Defining a Type 2.1 Channel” on page 140.

To define a channel-attached NCP:

1. Code a PCCU definition statement.
 - The CUADDR operand on the PCCU definition statement specifies the channel address of the channel adapter on the NCP that is used to perform the load operation. The operator can also specify the channel unit address on the VARY ACT operator command. VTAM identifies the channel link and link station by using the channel unit address.

- The LOADSTA and DUMPSTA operands on the PCCU definition statement can also be used to identify to VTAM the channel-link station to be used to load and dump the NCP.

Following is a sample PCCU definition statement for the channel-attached NCP in Figure 51:

```
PCCU116 PCCU CUADDR=016,          3720 IPL ADDRESS
           DUMPDS=NCPDUMP,        NCP DUMP DATA SET DDNAME
           MDUMPDS=MOSSDUMP,      MOSS DUMP DATA SET DDNAME
           CDUMPDS=CSPDUMP,      CSP DUMP DATA SET DDNAME
           VFYLM=YES,            NOTIFY OPERATOR IF WRONG LOAD MODULE
           VFYC=YES,            VERIFY LOAD MODULE AND RRT MATCH

:
SUBAREA=116          OWNING SSCP SUBAREA
```

In Figure 51, VTAM uses the CUADDR operand from the PCCU definition statement (016) to define the channel link (016-L) and channel link station (016-S). VTAM will load the 3720 using channel link 016-L.

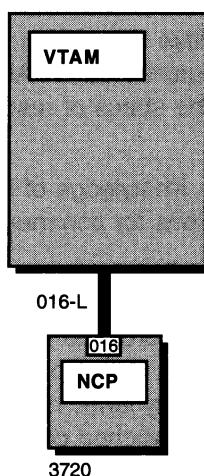


Figure 51. Channel-Attached 3720 Communication Controller

VTAM chooses default link stations that are adjacent to the NCP and ready for the dump operation. Preference is given to channel link stations.

If a channel-attachment major node and its link and link station minor nodes are active when this NCP is activated, use the channel-link station name that is defined by the PU definition statement in the channel-attachment major node for the DUMPSTA, LOADSTA, and RNAME operands of PCCU.

2. Code a BUILD definition statement.

Following is an example of these operands on the BUILD definition statement:

```
BUILD BUILD BFRS=240,          SIZE OF NCP BUFFERS

:
CA=(TYPE5,TYPE5),            CA TYPES
DELAY=(.2,.2),              CA ATTENTION DELAYS
NCPCA=(ACTIVE,ACTIVE),      CA STATUS
TIMEOUT=(30,30)             CA ANS - HOST FAILURE
```

One of the channel links is used to load the NCP.

3. You can also define channel links using the GROUP, LINE, and PU definition statements of the NCP, for a 3745 and optionally for a 3725 and 3720. The information about the channel-link characteristics can be coded as operands on these definition statements, similar to those used on the BUILD definition statement (CA, NCPCA, DELAY, and TIMEOUT). Code a LINE and a PU definition statement for each channel adapter. The names used to identify the channel link and link station to VTAM are the names specified on the LINE and PU definition statements, respectively.

Following is an example of how channel links can be defined using the GROUP, LINE, and PU definition statements of the NCP:

```

N111CA  GROUP LNCTL=CA           CHANNEL ADAPTER DEFINITIONS
L5CA05  LINE  ADDRESS=00,        CHANNEL ADAPTER 5 RELATIVE ADDRESS
                                CASDL=125,    CHAN ADAPTER SLOWDOWN DELAY
                                NPACOLL=YES,   NPA COLLECTION

:
                                CA=TYPE6,     CHANNEL ADAPTER TYPE
                                DELAY=.1,      CHANNEL ATTENTION DELAY
                                NCPCA=ACTIVE,  CHANNEL ADAPTER STATUS
                                TIMEOUT=180,   CHANNEL ADAPTER ANS - HOST FAILURE
                                TRANSFR=20,    MAXIMUM DATA TRANSFER SIZE
                                ISTATUS=ACTIVE  VTAM STATUS

*
P5CA05  PU    PUTYPE=5,          VTAM HOST CONNECTION - PUTYPE5
                                TGN=1         CHANNEL-LINK TRANSMISSION GROUP NUMBER

```

Using Parallel Transmission Groups between VTAM and a Channel-Attached NCP

A transmission group is one or more physical links connecting two subareas. These physical links are viewed as a single logical connection between the two subareas.

VTAM, with NCP V5R3 or higher for the 3745 and 3720, allows you to have multiple transmission groups between VTAM and a channel-attached NCP subarea node. You can add up to 16 transmission groups using up to 16 channel links between the same VTAM and NCP subarea nodes. However, because only single-link transmission groups are supported, several channel links between VTAM and NCP cannot be defined as a multilink transmission group.

To define a parallel transmission group:

1. Code transmission group numbers (TGN1, TGN3, TGN255) on the PU definition statement for each channel-link station.

In the following LINE and PU definition statements for the NCP, each channel link is defined as a separate transmission group (TGN1, TGN3, TGN255) by coding the TGN operand on the PU definition statement. However, the transmission groups TG7 and TG8 between the two NCPs can be comprised of multiple SDLC links.

N111CA	GROUP LNCTL=CA	CHANNEL ADAPTER DEFINITIONS
L5CA05	LINE ADDRESS=00,	CHANNEL ADAPTER 5 RELATIVE ADDRESS
:		
	CA=TYPE6,	CHANNEL ADAPTER TYPE
	ISTATUS=ACTIVE	VTAM STATUS
*		
P5CA05	PU PUTYPE=5,	VTAM HOST CONNECTION – PUTYPE5
	TGN=1	CHANNEL-LINK TRANSMISSION GROUP NUMBER
*		
L6CA06	LINE ADDRESS=01,	CHANNEL ADAPTER 6 RELATIVE ADDRESS
:		
	CA=TYPE6,	CHANNEL ADAPTER TYPE
	ISTATUS=ACTIVE	VTAM STATUS
*		
P6CA06	PU PUTYPE=5,	VTAM HOST CONNECTION – PUTYPE5
	TGN=3	CHANNEL-LINK TRANSMISSION GROUP NUMBER
*		
L7CA07	LINE ADDRESS=02,	CHANNEL ADAPTER 7 RELATIVE ADDRESS
:		
	CA=TYPE6,	CHANNEL ADAPTER TYPE
	ISTATUS=ACTIVE	VTAM STATUS
*		
P7CA07	PU PUTYPE=5,	VTAM HOST CONNECTION – PUTYPE5
	TGN=255	CHANNEL-LINK TRANSMISSION GROUP NUMBER

2. Code a transmission group number on the PCCU definition statement in the NCP major node or on the PU definition statement in a VTAM channel-attachment major node.

If a TG number is specified on the VTAM definition statement (PCCU or PU), it must match the corresponding TG number specification on the PU definition statement in the NCP for that channel link-station. Otherwise, code TGN=ANY, which allows the TG number to be assigned by the adjacent NCP, but the NCP subarea node must then specify a TG number. The transmission group number used is the TGN value that is coded on the PU definition statement in the NCP and the TG number that VTAM receives in the XID exchange during activation of the channel-link.

Transmission group number on the PCCU definition statement

One of the channel attachments must be used to load and activate the NCP from the VTAM host. The transmission group number that is used for that channel attachment can be specified on the PCCU definition statement.

Transmission group number on the PU definition statement

The channel links and link stations between VTAM and a channel-attached NCP must be defined in a channel-attachment major node.

3. Code PATH definition statements to define different routes using different channel links between the same VTAM and NCP subarea nodes. Although you can define up to 16 explicit routes between VTAM and a channel-attached adjacent NCP, the routing capability between these two subareas can be limited by the number of virtual routes that can be defined. The maximum is eight virtual routes using three different transmission priorities for a total of 24

potential routes. For information about how this affects network routing, see “Network Routing for Subarea Nodes” on page 151.

Following are two examples of how parallel transmission groups are defined:

- Figure 52 illustrates a possible configuration of parallel transmission groups.

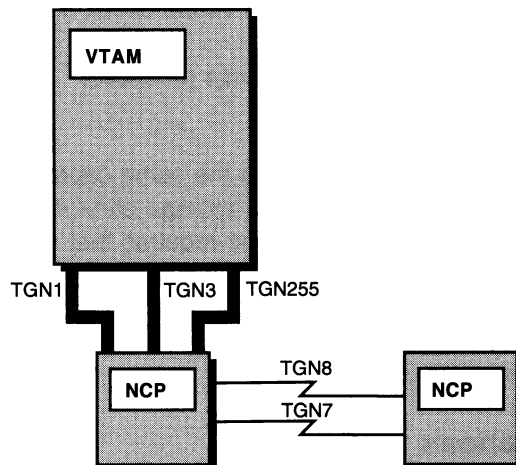


Figure 52. Parallel Transmission Groups in Single-Domain Environment

In Figure 52, the channel link station TGN1 can be used to load and activate the channel-attached NCP. In this case, the PCCU definition statement is used to specify the TG number (default is 1). The other two channel-link stations, TGN3 and TGN255, can be defined using a channel-attachment major node for each. In these two cases, the PU definition statement for each of the channel-link stations specifies the transmission group number. Because VTAM does not support the capability of defining multiple channel links as a single transmission group, each channel link in the NCP must be defined as a separate transmission group.

- The following is a PCCU definition statement:

```

PCCU116 PCCU AUTODMP=NO,      AUTOMATIC DUMP OPTION
          CUADDR=016,         3745 IPL ADDRESS

:
          TGN=ANY,            USE TGN OF NCP FROM XID

:
          SUBAREA=116         OWNING SSCP SUBAREA
  
```

In this example, the transmission group specification is TGN=ANY. Therefore, when contacted over the channel link identified by the CUADDR operand, VTAM uses the TG number that is specified on the PU definition statement for the channel-link station in the NCP. Because single VTAM now supports multiple channel attachments to one NCP, each channel attachment must use a different TG number so that the PATH statements between the subarea nodes can identify different network routes. Other channel attachments to the same NCP can be defined using the channel-attachment major node specification. In this case, the TG number specification is not on the PCCU definition statement but on the PU definition statement in the channel-attachment major node that represents that channel-link station.

Link-Attached NCP

A link-attached NCP can use any of the following connections:

- Nonswitched SDLC lines
- Switched SDLC lines
- Token-ring networks
- **VM, VSE** X.25 permanent or switched virtual circuit (PVC or SVC)
- **VSE** X.21 switched or nonswitched
- **VM, VSE** V.25 bis for the ES/9000 series processor
- 3172 interconnect controller.

The use of a switched SDLC link provides the same capabilities as a nonswitched link except that a static dump of controller storage cannot be transferred directly to the host. The use of the other connections requires that an initial load module be available on the controller disk. However, after the initial load module is activated by VTAM, subsequent load modules can be transferred over the token-ring, X.21 switched, or X.25 switched virtual circuit (SVC) connections, using 3720 and 3745 multiple load module support.

Recovery Operations

If a functional NCP is not present in the controller, the controller does not accept the transfer of load modules over these connections. Perform a recovery operation using one of the following methods:

- Reload NCP over a switched SDLC link.
- Using a diskette procedure, transfer an initial load module onto the controller disk; then reload the controller.

The capability to load and activate or dump a link-attached NCP over a token-ring, X.21, or an X.25 switched virtual circuit (SVC) connection, requires additional planning. For more information about this, refer to *Planning for NetView, NCP, and VTAM*.

Defining a Link-Attached NCP

Figure 53 illustrates a link-attached NCP (NCP2). The connection between a channel-attached and a link-attached NCP is defined in each of the NCP subarea nodes. You define each end of the connection in each NCP using a LINE definition statement for the link and a PU definition statement for the link-station.

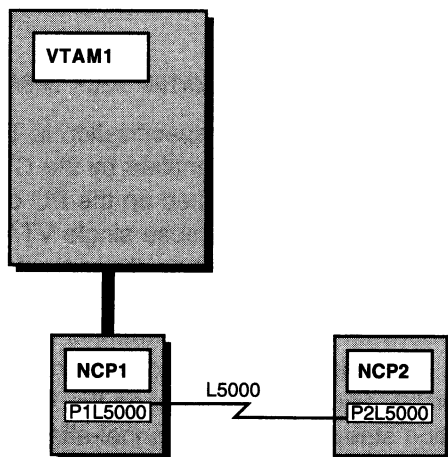


Figure 53. Defining, Loading, and Activating a Link-Attached NCP

The following example illustrates the definitions (at the channel-attached NCP, NCP1) for the link and link-station when using a nonswitched SDLC link to the link-attached NCP subarea node:

```

*-----
* 256 KILOBYTE LINK
*-----
G5INN      GROUP  ACTIVTO=120,          ACTIVATE TIME OUT VALUE
              DIAL=NO,                  NONSWITCHED LINK
              LNCTL=SDLC,                SDLC LINE CONTROL
              REPLYTO=(10.0,15.0),       REPLAY TIMEOUT VALUES

:

              ISTATUS=ACTIVE            VTAM STATUS
*
L5000      LINE  ADDRESS=(000,FULL),     LINE ADDRESS - FULL DUPLEX
              SPEED=245760,              LINK SPEED
              HISPEED=YES,                HIGH SPEED LINK
              MONLINK=YES,                SDLC MONITOR MODE ON
              IPL=YES,                    ACCEPT IPL REQUEST

:

              MODULO=128,                 MODULO COUNT = 128
              PAUSE=(1.0,1.0),            PRIMARY AND SECONDARY PAUSE
              RETRIES=(3,12,5),           RETRY VALUES
              SDLCST=(PRIM,SEC),          PRIMARY AND SECONDARY CHARACTERISTICS
              ISTATUS=ACTIVE              VTAM STATUS
*
P1L5000    PU    ANS=CONTINUE,            AUTOMATIC NETWORK SHUTDOWN
              MAXOUT=127,                 FRAMES PRIOR TO LINK-LEVEL REPLY

:

              PUTYPE=4,                   TYPE 4 - NCP CONNECTION
              TGN=7,                       TRANSMISSION GROUP NUMBER
              ISTATUS=ACTIVE               VTAM STATUS

```

To load and activate the link-attached NCP:

1. Specify the link-station name of the channel-attached NCP (in the example, P1L5000) in the RNAME operand of either the PCCU definition statement for the link-attached NCP (NCP2) or the VARY ACT command.

On the PCCU Definition Statement for NCP2:

```

PCCU  SUBAREA=1,
:
      LOADSTA=P1L5000,
      RNAME=P1L5000,
:

```

On the VARY ACT Operator Command:

```
VARY ACT,ID=NCP2,LOAD=YES,RNAME=P1L5000
```

2. Define the characteristics of the link. The NCP subarea nodes exchange information over the link using the XID process. The subarea node with the higher subarea number performs the role of the primary and performs the polling unless this role relationship is predefined. The characteristics associated with

the link, depending on primary or secondary role, must be defined using the SDLCST and GROUP definition statements in the NCP subarea nodes.

3. **VM,VSE** Use a channel-attachment major node to define SDLC, LAN, or X.25 links to a communication controller.

Cross-subarea links can be directly or indirectly activated and deactivated. For information about operating with links, see "Activating Links" on page 375 and "Deactivating Links" on page 381.

Enhanced Remote Controller Support

Enhanced remote controller support enables you to load, activate, and dump NCP using a switched SDLC link, including a token-ring connection, an X.25 switched virtual circuit (SVC), and an X.21 switched link. However, the token-ring, X.25 SVC, and X.21 switched connections require an NCP to be operational in the communication controller or available on the controller disk.

Following is a sample switched major node for enhanced remote controller support:

NCPSW111	VBUILD TYPE=SWNET,	SWITCHED MAJOR NODE
	MAXGRP=5,	MAXIMUM PATH GROUPS
:		
	MAXNO=5	MAXIMUM TELEPHONE NUMBERS
*		
P5001T4	PU ADDR=C1,	PU TYPE 4 STATION ADDRESS
	MAXDATA=4096,	MAXIMUM DATA TRANSFER
	NETID=USIBMCSS,	NETWORK IDENTIFIER
	SUBAREA=112,	SUBAREA NUMBER
	IDNUM=11201,	UNIQUE IDNUM
	PUTYPE=4,	PU TYPE 4
	TGN=8,	TRANSMISSION GROUP NUMBER
:		
	MAXPATH=2	MAXIMUM DIAL OUT PATHS
*		
PATH	DIALNO=1234567,	TELEPHONE DIAL NUMBER
	GID=2,	GROUP IDENTIFIER
	PID=2,	PATH IDENTIFIER
	REDIAL=4,	RETRY DIAL 4 TIMES
:		
	GRPNM=G6INN	GROUP NAME IN NCP

To use enhanced remote controller support:

1. Define the switched physical connection by coding the NCP LINE and PU definition statements with the DIAL operand specified as YES.
2. Define the corresponding logical definition to VTAM by coding a PU definition statement for a type 4 PU in a switched major node.
3. Because a subarea physical unit can have multiple connections, code a connection identifier (IDNUM) to distinguish the multiple PU definition statements that can apply to a subarea physical unit. The value of the IDNUM operand on the PU definition statement must match the value of the IDNUM operand on the SWMMT definition statement for the remote communication controller; and when the IDNUM operand is concatenated with the subarea number for the

NCP, it must be unique within the network. Therefore, you can use the same IDNUM to represent connection endpoints in different subareas. If IDNUM is duplicated within the network, a connection can be established with the wrong subarea, which may result in a connection failure or other errors later in the process. IDBLK is not used by VTAM and is set to a value of X'FFF' in the station identifier that is used for the XID exchange during the dial procedure.

4. Code a PATH definition statement so that VTAM performs a dial-out operation on the switched link and contacts the remote device (PUTYPE=4). The PATH definition statement also specifies the group name (GRPNM) of the group of switched lines in the NCP over which the connection can be made.
5. Make sure the value of the NETID operand on the PU definition statement in the switched or model major node matches the NETID operand on the SWMMT definition statement for the corresponding switched line in the remote communication controller.

Note: The IDNUM and NETID operands are maintained within the NCP load module so an NCP generation is required to change these values.

6. For security reasons, you might want to code a password on the PU definition statement using the PRTCT operand. If the check fails, the connection fails.

Note: In a single-domain network where a single VTAM and NCP control both link stations, VTAM does not support call security verification. (That is, if you code either the MONLINK=YES or MONLINK=CONT operand for the switched line in the link-attached NCP, VTAM does not support call security verification.)

NCP Switched SDLC Connections

For the following discussion, normal dial-in refers to a call originating from a physical unit; automatic dial-out refers to a call established by VTAM using a communication controller. Manual dial-out refers to a VTAM-operator-assisted call from the communication controller or communication adapter to the physical unit.

Dial-Out Connections: When an application program requests a session, VTAM selects a dial-out line, unless the application program specifies the dial-out line. Depending on the capabilities of the switched line, the dial-out call can be handled automatically (done entirely by VTAM or by VTAM and the NCP) or manually (done partly by the VTAM operator).

Dial-out requests (except for X.21 Direct Call) can be placed on one of a number of switched paths to a physical unit as defined within a switched major node. The dial-out path assignment for switched physical units can be controlled by making switched paths usable or not usable with the VARY PATH command. This command can be used for individual switched paths by specifying the path identifier (PID), or for a group of switched paths by specifying the group identifier (GID). For example, all switched paths that use Wide Area Telephone Service (WATS) lines for dial-out connections can be associated with a GID in the switched major node definition, and can be made usable or not usable as a unit.

Automatic dial-out

VM, VSE Automatic dial-out refers to a call established by VTAM using a communication-adapter autocal unit. VTAM sends the telephone number to the communication controller or communication adapter. The connection to the physical unit is established with the communication-controller autocal unit.

VSE For automatic dial-out to an X.21 network, an autocal unit is not required and, if X.21 Direct Call is being used (communication controller only), the telephone number is not sent. When VTAM is notified that the connection is complete, the identification of the physical unit is verified. From this point, the process continues as for a dial-in connection.

Manual dial-out

For a manual dial-out call, VTAM displays a message with the line and telephone number to use. The VTAM operator manually calls the physical unit. VTAM is notified when the switched connection is complete and then verifies the identity of the physical unit. From this point, the process continues as for a dial-in connection.

If the VTAM operator is unable to complete a manual dial-out call, the VARY INOP command can be used to specify whether VTAM is to select an alternate line or terminate the dial-out operation.

Dial-In Connections: For normal dial-in operation, the terminal operator calls the communication controller or communication adapter, and VTAM is notified of the incoming call. The NCP then requests that the physical unit identify itself. When the physical unit responds, VTAM locates the proper physical unit definition in an active switched major node. After VTAM completes activation of the physical unit and its subordinate logical units, sessions involving the logical units can be initiated as if the line were nonswitched.

Switched lines with dial-in capability can allow or disallow incoming calls. For an incoming call to be allowed, the line must be in answer mode. The mode setting can be specified in the definition statement for the line and can be changed with the ANS operand of the VARY ACT command when activating the line, or with the VARY ANS command after the line is active.

V.25 Bis Protocol (VM, VSE): You can connect peripheral nodes to VTAM in an ES/9000 series processor using a V.25 bis protocol modem. To send V.25 bis commands to the modem, code the commands as character strings on the DIALNO operand of the PATH definition statement. You can code up to 32 characters, and VTAM will pass the commands unmodified to the modem.

Defining Dependent LUs Dynamically

To enable this facility, code the LUGROUP operand on the PU definition statement for the physical unit, and code an LU group major node. The LUGROUP operand on the PU definition statement specifies the name of the model LU definition group that VTAM uses when dynamically defining logical units. The LU group major node contains the model definition statements. Dynamic definitions for logical units are built using the model LU definitions contained in this major node.

Note: Any operands that you code on the PU definition statement will not sift down to dynamically created LU definition statements.

After you enable this facility, VTAM uses the IBM-supplied selection of definitions for dependent LUs (SDDL) exit routine to generate LU names for the dynamically built LU definition statements. The SDDL exit routine controls the selection of the model LU and the LU name. If you prefer to use your own naming convention or method of selecting an LU model, you can write your own exit routine. For informa-

tion on coding your own exit routine, see “Selection of Definitions for Dependent LUs (SDDL) Exit Routine” in *VTAM Customization*.

If you use the IBM-supplied SDDL exit routine, code the LUSEED and LUGROUP operands on the PU definition statement. The LUSEED operand provides a pattern name that is used with the SDDL exit routine to create a name for the dynamically created LU definition statements.

Figure 54 illustrates VTAM dynamically defining dependent logical units attached through a 3174 cluster controller.

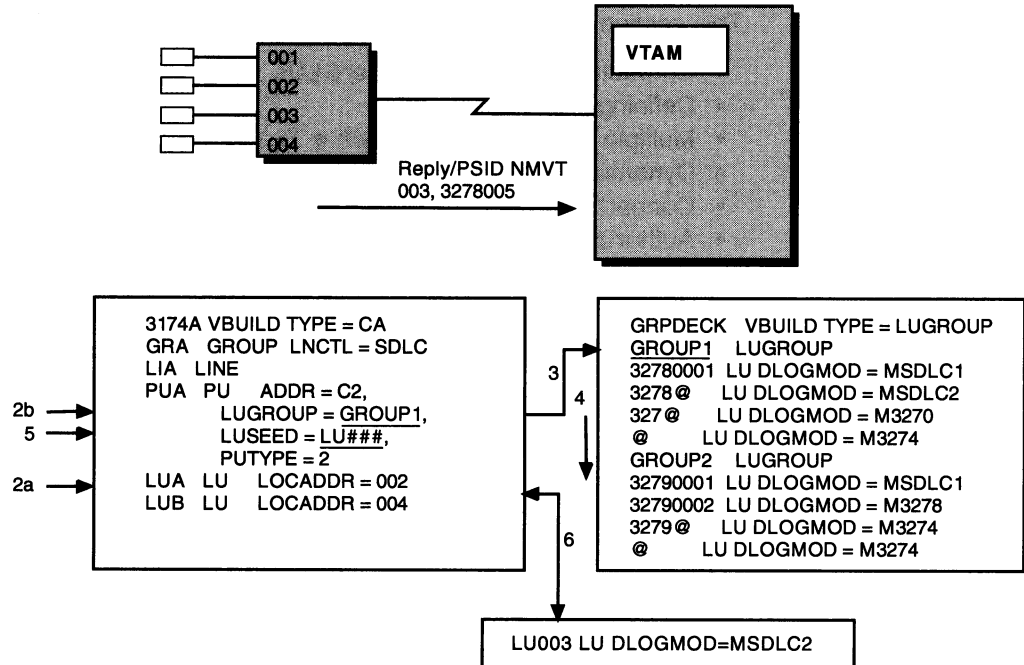


Figure 54. Definition Building for Dynamically Defined Dependent LUs

1. The logical unit at local address 003 powers on. The 3174 sends VTAM a Reply/PSID NMVT indicating a 3278 model 005 device has powered on at address 003.
2. VTAM uses the Reply/PSID NMVT to:
 - a. Check the 3174's PU definition statement for a definition of a logical unit at address 003
 - b. Determine if the 3174 supports dynamic definition of dependent logical units (LUGROUP operand is coded), if the 3174's PU definition statement has no definition of a logical unit at address 003.
3. VTAM matches the LU group name specified in the 3174's PU definition statement with an LU group name in the LUGROUP definition statement.
4. The default SDDL exit routine uses the seven-character machine type and model number as the model LU name. VTAM serially checks the LU group for a model LU statement that matches the model name supplied by the SDDL exit routine. In this example, VTAM finds the second definition statement (3278@). The characteristics specified in that statement are used.
5. To create the logical unit name, the default SDDL exit routine uses the value on the LUSEED operand. In this example, the three # signs specified on the LUSEED operand are replaced with the numeric local address. Therefore, the

name of the LU is LU003. For more options, see “LUSEED” in the *VTAM Resource Definition Reference*.

6. VTAM builds a dynamic definition for the logical unit. The dynamic definition is rebuilt if that device powers off and a new device powers on at the same local address. Also, the dynamic definition is rebuilt if the device powers off and powers back on after its LU group major node has been inactivated and then reactivated.

Independent LUs

This section contains the following subsections:

- Characteristics of Independent LUs
- Defining Independent LUs
- Multiple Connections between a Type 2.1 Node and a Subarea Node
- Dynamic Selection of Session Connections
- Connecting Independent LUs through NCPs
- Authorized Transmission Priority for LEN Connections
- Restrictions on Using Independent LUs.

Characteristics of Independent LUs

Following are some characteristics of independent LUs.

Adjacent Link Stations: Adjacent link stations are type 2.1 physical units (or type 4 or type 5 PUs that appear as type 2.1 PUs) through which independent logical units can access the network. There is no hierarchical relationship between an adjacent link station (PU) and an independent LU. For more information about adjacent link stations and how independent logical units can use them to access the network, see “Using CDRSC Definition Statements for Independent LUs” on page 222 , and “Multiple Connections between a Type 2.1 Node and a Subarea Node” on page 224 .

Session Capabilities: The session capabilities of an independent logical unit are significantly greater than those of dependent logical units. Independent logical units do not require SSCP-PU or SSCP-LU sessions. An independent logical unit can act as a primary logical unit and can both initiate and respond to session requests.

Extended BIND: Independent logical units support the extended BIND, which contains information necessary to activate a session, to make use of features such as adaptive session pacing, and to identify a particular session. Extended BINDs always contain a fully qualified procedure-correlation identifier (a PCID qualified by the name and network identifier of the PCID generator). A BIND without this information is a non-extended BIND. Extended BINDs can contain network-qualified names if the subarea components in the session establishment path support them. An extended BIND is sent only if the boundary function (if applicable) and the destination logical unit support extended BIND.

Multiple Sessions: An independent logical unit can also have multiple sessions with a single logical unit (parallel sessions) or with several different logical units. The independent logical unit can assume the role of primary or secondary for any one or all of the sessions.

If one or more sessions are required between logical units for application transaction program communication, the independent logical unit establishes the necessary session connections on behalf of those application transaction programs.

The BSBUF buffer pool is used to provide common storage for boundary type 2.1, type 2, and type 1 peripheral node session control blocks. Depending on how you code the buffer pool start option, the BSBUF buffer pool can expand to support more sessions and contract as less session control blocks are needed. Because independent logical units can have multiple sessions (including parallel sessions), you should determine the average and maximum number of sessions associated with independent logical units that are directly attached to that VTAM subarea node. You can then specify a value that prevents excessive buffer expansion and contraction.

Addressing: An independent logical unit in a type 2.1 peripheral node does not have a predefined address. Although VTAM definition statements or an NCP generation associates an independent LU with a network accessible unit, the address of that unit does not have to be coordinated with VTAM or NCP coding. For information on how NCP handles address assignments, see “Connecting Independent LUs through NCPs” on page 226.

Alias Name Translation and Independent LUs: Following are certain restrictions that apply to alias name translation and the independent LUs:

- If the PLU name received in the BIND from the LU is network-qualified, the SLU name must also be network-qualified.
- If the names are network-qualified, they must be the real names (unless they are USERVAR names), or the session setup fails.
- Concurrent sessions to destination resources with the same name in different networks cannot be established when using alias name translation. Concurrent sessions to destination resources with the same name in different networks can be established if the independent LU's host is using NQNMODE=NQNAME to define the destination resources. For more information on NQNMODE=NQNAME, see “NQNMODE” in the *VTAM Resource Definition Reference*.
- If the SLU name received in the BIND is network-qualified, and the SLU is from a different network than the PLU, the SLU's alias name in the PLU's network will be the same as the SLU's real name.

Dynamic Reconfiguration and Independent LUs: You can create and alter CDRSC major nodes as needed to effect the same level of support as dynamic reconfiguration provides. Because dynamic reconfiguration has meaning only if the independent LUs are predefined using LU definitions, it is no longer needed once you convert your independent LU definitions to CDRSCs.

Note: The ISTPDILU major node contains CDRSC definitions for independent LUs that are converted from LU definition statements by VTAM. The ISTPDILU major node is automatically activated by VTAM on initialization and cannot be activated or deactivated via operator commands.

When processing a dynamic reconfiguration definition that contains an independent LU definition to be added, VTAM converts that independent LU to a CDRSC and places it in the ISTPDILU major node. The PU named on the TO operand becomes an entry in that CDRSC's default adjacent link station list.

If you specify a PU on the FROM operand of the dynamic reconfiguration DELETE command on an independent LU, VTAM deletes the network address associated with that independent LU that was predefined for that independent LU. Additionally, the PU is removed from the default adjacent link station list if present.

To use the dynamic reconfiguration DELETE command on an independent LU, the independent LU must not have any sessions over the PU named on the FROM operand.

You cannot use the MODIFY DR command for independent logical units.

Defining Independent LUs

You can define independent logical units in the following ways:

- You can have your independent logical units defined dynamically by VTAM during session establishment. See “Dynamic Definition of Independent LUs” on page 221.
- You can define CDRSC definition statements to represent your independent logical units. See “Using CDRSC Definition Statements for Independent LUs” on page 222.
- You can use standard LU definition statements with LOCADDR=0 coded. These LU definitions are converted into CDRSC definitions by VTAM when the major node is activated. This method is not recommended other than as a migration tool. Using this method, you cannot warm start VTAM and have your independent logical units return to a before-failure condition. Independent logical units that are defined this way are converted to CDRSCs by VTAM and placed in the ISTDILU major node. If a new resource is detected that matches an existing CDRSC, then the new resource definition is integrated with the existing definition, and the resulting resource resides in the ISTDILU major node. See “Automatic Conversion of Independent LU Definition Statements” on page 223.

For information on how VTAM handles multiple definitions of one independent LU resource, see “Rules for Multiple Definition of Resources” in the *VTAM Resource Definition Reference*.

Migration: It is recommended that you remove pre-defined independent LU definitions from your device-type major nodes and either let VTAM dynamically define CDRSC definitions for the independent LUs on a session-request basis or create CDRSC major nodes. If you decide to code CDRSC major nodes for your independent LUs and you previously had them defined to the NCP, you should regenerate your NCP to avoid any address mismatch problems.

One of the advantages of removing existing LU definition statements for independent logical units (and using either dynamic definition of independent logical units or coding CDRSC definition statements) is that predefined, NCP generated addresses are not used by the independent logical unit. Addresses are requested as needed and are returned when no longer needed.

You can use multiple connections and the dynamic selection of session connections without dynamically defining your independent LUs as CDRSCs.

Note: If you have session takeover operations that rely on VTAM creating substitute LUs, these takeover operations can fail.

Dynamic Definition of Independent LUs

Independent LUs attaching to VTAM through a type 2.1 PU can be dynamically defined as cross-domain resources. No system definition of the independent LU is required.

If the independent logical unit initiates the session, the adjacent link station (ALS) through which the independent logical unit contacts the network does not have to be predefined to VTAM. For example, if the independent logical unit is attached through a token-ring network, then the adjacent link station would be specified in the form of the medium access control address (MACADDR) of the host. VTAM discovers the resource during session startup and dynamically creates a CDRSC to represent the logical unit.

If the independent logical unit is the destination logical unit rather than the originating logical unit, and the destination logical unit is located on a LEN node, then you can dynamically define the independent logical unit using the adjacent link station (ALS) selection function of the session management exit routine. (An independent logical unit that is the destination logical unit on an APPN node has no such restriction.) Using this exit routine, you can provide VTAM with a method of contacting the independent logical unit. For information on the ALS selection function, see “Dynamic Selection of Session Connections” on page 225. For information on writing a session management exit routine, see “Session Management Exit Routine” in *VTAM Customization*.

Note: The ALS selection function of the session management exit routine is not required if the destination logical unit is a LEN CP independent LU. Use the CPCDRSC=YES start option for this situation. If a session request is made to a LEN CP independent LU and CPCDRSC=YES is coded, VTAM builds a dynamic CDRSC to represent the LEN CP independent LU. As a result, other network resources will be able to establish sessions with the LEN CP independent LU before it has attempted to establish any sessions.

To have your independent logical units dynamically defined:

1. Code DYNLU=YES in the start list, or use it on the START command when starting VTAM; or code DYNLU=YES on all PU definition statements that represent the adjacent link stations that the independent logical units use to connect to the host. If you code DYNLU=YES when starting VTAM (either in the start list or on the START command), then you can restrict dynamically defined devices from using specific physical units by coding DYNLU=NO on the PU definition statements. If a PU has DYNLU=NO specified, VTAM does not select this PU as the adjacent link station.

Coding DYNLU=YES enables VTAM to learn about the independent LU during session activation and dynamically define representations of the LU.

2. Set the CDRSCTI start option, if you want to change the amount of time that the dynamic definitions of independent LUs are retained after session termination (default = 8 minutes).
3. The default for the CDRDYN start option is YES, enabling dynamic CDRSC definition. The CDRDYN start option overrides the CDRDYN operand on the host CDRM definition statement.
4. Code the CPCDRSC=YES start option in a host that is connected to a LEN CP to allow dynamic definition of the LEN CP independent LU when it is the desti-

nation LU. DYNLU still controls dynamic definition of all independent LUs (including LEN CPs) when they are the originating logical unit.

When an independent LU is dynamically created by VTAM, the following defaults are used:

- ENCR=NONE
- RESSCB=0
- EAS=256
- MODETAB=ISTINCLM (IBM-supplied logon mode table).

The PACING and VPACING operands specified on the PU definition statement associated with the session are sifted down to the CDRSC. The default logon mode table (MODETAB operand) is the IBM-supplied default table, ISTINCLM. The logon mode for each session is extracted from the BIND request which is passed to VTAM.

If the adjacent link station that the independent LU is using is being dynamically defined using dynamic switched definitions, you need to ensure that you have an entry for the mode you want to use in the default logon mode table, ISTINCLM. In this case, VTAM checks only ISTINCLM for the mode entry. If the mode entry is not in ISTINCLM, the session fails. After the session is started, you can change the logon mode table using the MODIFY TABLE command.

Dynamic CDRSCs are created and maintained in the ISTCDRDY major node. ISTCDRDY is activated automatically during VTAM initialization, and deactivated automatically during VTAM termination. The operator can deactivate ISTCDRDY with the VARY INACT command, in which case all dynamically defined CDRSCs are also deactivated and the dynamic CDRSC function is disabled. In addition, all sessions involving dynamically created CDRSCs (including CP-CP sessions with this host, if the partner CP was dynamically defined) are terminated.

For information on how VTAM handles multiple definitions of one independent LU resource, see “Rules for Multiple Definition of Resources” in the *VTAM Resource Definition Reference*.

You can use dynamic definition of independent logical units with other functions, such as multiple connections to a subarea, casual connection, or APPN.

Using CDRSC Definition Statements for Independent LUs

You can code CDRSC definition statements for your independent logical units, and specify the adjacent link stations (physical units) that VTAM uses to contact the independent logical unit. You can specify the adjacent link stations either by using the ALSLIST operand on the CDRSC definition statement, or by using the adjacent link station selection function of the session management exit routine. For more information on using the exit routine, see “Dynamic Selection of Session Connections” on page 225.

The following sample shows an independent logical unit defined using a CDRSC definition statement:

```

CDRSC      VBUILD TYPE=CDRSC
VRCDRSC    CDRSC ALSLIST=(SWPUAI06,SWPUA107)  ADJACENT LINK STATION NAMES
                                           USE ALS LIST ONLY
                                           LOGAPPL=APPL1,      AUTOMATIC LOGON
                                           DLOGMOD=DSILGMOD,    LOGON MODE TABLE ENTRY
                                           RESSCB=3            SESSION CONTROL BLOCKS

```

ALSLIST=*puname* indicates to VTAM that this resource is an independent logical unit defined as a CDRSC. The ALSLIST operand can list one or more adjacent link stations that VTAM uses. The ALSLIST operand can also include an ISTAPNPU entry, a reserved keyword that represents any APPN capable PU. This entry can be used instead of listing each APPN adjacent link station. For information on controlling connection to independent LUs that are accessible over both LEN and APPN connections, see “Establishing and Controlling Sessions” on page 512.

You can add and delete adjacent link stations in the list of adjacent link stations using the MODIFY ALSLIST command. For more information on this command, see “MODIFY ALSLIST Command” in *VTAM Operation*.

You can use the adjacent link station selection function of the session management exit routine to dynamically add other adjacent link stations to the list and to perform other adjacent link station selection functions. For details, see “Session Management Exit Routine” in *VTAM Customization*.

You can use the ALSREQ operand or start option to control which adjacent link stations can be used for session traffic to independent logical units. The ALSREQ operand indicates whether this independent logical unit can establish sessions with an adjacent link station that is not defined by the ALSLIST operand. ALSREQ=YES means that a predefined adjacent link station is required for session establishment. If you code ALSREQ=NO, any adjacent link station can receive a session request for this logical unit.

Notes:

1. Even when ALSREQ=YES is specified, if any APPN adjacent link station name (or ISTAPNPU) exists in the ALS list for the CDRSC, any APPN adjacent link station can receive a session request for this logical unit.
2. Any operand normally coded on an LU definition statement can be coded on a CDRSC definition statement when defining an independent logical unit. These operands only have an effect when the CDRSC is an independent logical unit.

Automatic Conversion of Independent LU Definition Statements

The ISTPDILU major node contains CDRSC definitions for independent LUs that are converted by VTAM from LU definition statements. In converting the independent LU to a CDRSC, VTAM removes the implied hierarchy of the independent LU to a type 2.1 physical unit.

The converted CDRSCs can be used with multiple connections between a type 2.1 node and a subarea node. For this reason, the individual CDRSC minor nodes that are converted from LU definitions are not freed when the NCP (or other major node) in which they were originally defined is deactivated.

However, you can delete individual CDRSC minor nodes without deleting the entire major node. You can use the DELETE operand on the VARY INACT command on predefined CDRSC minor nodes.

When the LU definition statement is converted, the resource is represented as a CDRSC. This means that if the OWNER operand is specified on the LINE definition statement where the independent LU is predefined, it has no effect on the independent LU being converted to a CDRSC. The independent LU is converted to a CDRSC, but the PU under which it is predefined cannot be used as a session path until it is acquired by that host.

Multiple Connections between a Type 2.1 Node and a Subarea Node

An independent LU can connect to the subarea network through only a type 2.1 PU. A type 2.1 PU can be connected to the subarea network through a type 4 or type 5 PU. The type 2.1 PU connections are called adjacent link stations. An independent LU can connect to multiple adjacent link stations. Thus, the independent LU can have multiple, simultaneous connections to the VTAM host.

If one connection fails, only the sessions using that connection must be re-established; the other sessions are not affected. The independent LU can then use an alternative adjacent link station for any subsequent sessions. You can restart any sessions terminated by the first adjacent link station failure and connect the independent LU through another adjacent link station.

Some of the independent LU's session paths might involve type 2.1 nodes connected to a VTAM other than the VTAM of the desired session partner. For these sessions, the intermediate VTAM views the independent LU as a CDRSC. The CDRSC that represents the independent logical unit can also have other concurrent sessions that use PU type 2.1 nodes connected to the VTAM of the desired session partner.

If an independent logical unit is connected to VTAM by only LEN links and two application programs try to contact it, VTAM provides the first available adjacent link station specified on the ALSLIST operand for both the application programs. Both application programs would then have parallel sessions through the same adjacent link station. If you want the application programs to use different adjacent link stations, use the adjacent link station selection function of the session management exit routine. For information on coding this exit routine, see "Session Management Exit Routine" in *VTAM Customization*.

Figure 55 shows how an independent LU can connect to VTAM with multiple connections.

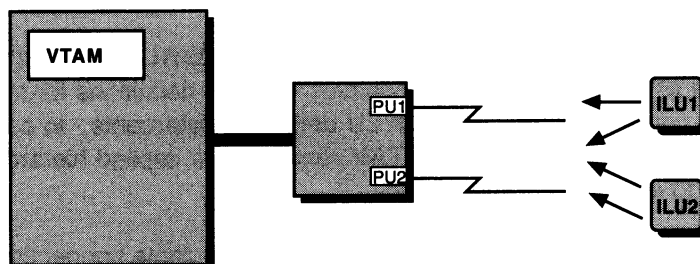


Figure 55. Independent LU with Multiple Connections to VTAM

Following are sample definition statements to define the configuration in Figure 55. Note that you can support independent logical units as either destination logical units, originating logical units, or both:

- For independent logical units as originating logical units, code DYNLU=YES in your start list. VTAM determines the adjacent link stations dynamically.
- For independent logical units as destination logical units, code CDRSC minor nodes in a CDRSC major node. If the independent logical units are connected to VTAM by only LEN connections, include the ALSLIST operand.

Start list (ATCSTRxx)

If the independent logical units are originating logical units, code the start option for dynamic definition of independent logical units (DYNLU=YES).

NCP major node

Code the following (do not code LU definition statements):

```

:
:
:      LUDRPOOL NUMILU=2      NUMBER OF INDEPENDENT LUS
LGROUP GROUP DIAL=NO        LEASED LINE
LLINE  LINE
:
:      PU1      PU
:
:      SGROUP   GROUP DIAL=YES      SWITCHED LINE
SLINE  LINE
:
:      PU2      PU
:
:

```

Switched major node

Code a switched major node:

```

VBUILD  TYPE=SWNET
PUA      PU      ADDR=03,          STATION ADDRESS
          CPNAME=A500N,          CONTROL POINT NAME (TYPE 2.1)
          DISCNT=YES,            DISCONNECT FACILITY
          MAXPATH=5,             NUMBER OF DIAL-OUT PATHS
          PUTYPE=2                PU TYPE 2.1

```

Note: You can code the DYNLU operand on the PU statement rather than in the start list.

CDRSC major node

If the independent logical units are destination logical units, define your independent LUs as CDRSCs:

```

VBUILD  TYPE=CDRSC
ILU1     CDRSC ALSLIST=(PU1,PU2),  ADJACENT LINK STATION NAMES
          ALSREQ=YES               USE ALS LIST ONLY
ILU2     CDRSC ALSLIST=(PU1,PU2),  ADJACENT LINK STATION NAMES
          ALSREQ=YES               USE ALS LIST ONLY

```

Dynamic Selection of Session Connections

If an independent LU has an existing leased or switched LEN connection (and no APPN connection), VTAM can dynamically select session paths for outbound sessions from the boundary function to the independent LU. VTAM attempts to use leased connections first, then tries active switched connections. If the first adjacent link station in the adjacent link station list is not available, VTAM tries to use an alternate connection from a default adjacent link station list that VTAM creates. When the first adjacent link station (the primary connection) does become available,

VTAM uses that adjacent link station for any new sessions. However, sessions already allocated to the alternate adjacent link station do not change.

To control selection of an alternate session path when an independent logical unit is connected to VTAM by only LEN links, use the ALS selection function of the session management exit routine. VTAM then establishes the path between the independent LU and the boundary function; this continues to be the path for the remainder of that session. You can select a different adjacent link station for each session.

Using the ALSREQ start option, you can limit the adjacent link stations that an independent LU can use.

You can code the session management exit routine to choose a PU from the default list or another PU by overriding the default PU list. For information on writing a session management exit routine, see “Session Management Exit Routine” in *VTAM Customization*.

The default adjacent link station chosen must:

- Act as a type 2.1 PU
- Reside in a major node that supports independent LUs
- Be active, if it is nonswitched
- Be active, connectable, or pending dial if it is switched.

Connecting Independent LUs through NCPs

You should consider the following when connecting independent LUs through NCPs. For further information, refer to the *NCP, SSP, and EP Resource Definition Guide*.

LU Address Pool: NCP must generate a pool of network addresses to establish switched connections. The address pool is generated in NCP using the LUDRPOOL definition statement. The NUMILU operand on the LUDRPOOL definition statement specifies the number of independent logical unit switched connections. You need to know the maximum number of concurrent independent LU switched connections so that you can reserve the appropriate number of independent logical unit control blocks in the NCP.

Primary LU Address Pool: When an independent LU acts as the primary logical unit by initiating a session, VTAM and NCP must assign another network address to the PLU. The AUXADDR operand on the BUILD definition statement reserves this address pool. Because this is a predefined pool, you should know the maximum number of concurrent PLU sessions that can be established by all independent logical units that are attached to that NCP. Otherwise, session establishment for an independent LU, acting as the primary logical unit, fails when the pool is exhausted.

Session Information Pool: You need to know the maximum number of sessions associated with all of the type 2.1 peripheral nodes attached to the NCP subarea node. NCP reserves a pool of control blocks for maintaining information about these sessions. The pool of control blocks is established during the NCP generation process using the ADDSESS operand on the BUILD definition statement. If this pool is exhausted in the NCP, session establishment fails because the pool is fixed and cannot expand dynamically.

Reserving Session Control Blocks: You can also reserve session control blocks for each independent logical unit by specifying a value for the RESSCB operand on the LU definition statement.

Limiting Sessions: To limit the number of sessions a single independent LU can establish per link station, you can use the MAXSESS operand on the:

- LU definition statement in an NCP major node, switched major node, or dynamic reconfiguration file
- PU definition statement in a model major node
- CDRSC definition statement in a cross-domain resource major node.

This prevents a single independent LU from using all of the session control blocks generated in the NCP.

Improving Session Establishment Performance: Use of the EAS operand can improve the performance of VTAM in establishing independent logical unit sessions. The EAS operand on the LU or CDRSC definition statement estimates the number of concurrent sessions for the logical unit. For performance reasons, if you anticipate that a particular logical unit might have more than 256 concurrent sessions (the EAS default), increase this number to the nearest multiple of 256.

Authorized Transmission Priority for LEN Connections

Normally it is desirable for a session between two independent LUs through a subarea network to use the same transmission priority for both type 2.1 LEN connections (entry and exit). By default, when VTAM receives a BIND over a type 2.1 LEN connection, the transmission priority field received on the BFINIT is passed back on the BFCINIT and through the subarea network on the BIND. This allows this same transmission priority to be used for the session by the type 2.1 LEN connection on exit from the subarea network.

To have VTAM ignore the transmission priority specified on the BFINIT, use the AUTHLEN=NO start option or the AUTHLEN=NO operand on a particular PU definition statement (or on a GROUP or LINE statement to sift down to the PU). VTAM then sets the transmission priority field in the BFCINIT to B'01' (medium priority), regardless of the transmission priority specified on the initial BFINIT.

Restrictions on Using Independent LUs

Following are some restrictions that apply to independent logical units:

- An independent LU cannot be the destination logical unit for an automatic logon session (for example, LOGAPPL).
- A dependent LU can have only one active LU-LU session at a time. Thus, there is no session with the controlling application program when the LU is in session with another application program. For independent LUs, this is not the case. VTAM initiates a session between an LU and its controlling application program, even though there are already active LU-LU sessions. Each time that a session with a PLU that is not the controlling application program ends, and the independent LU does not already have a session with the controlling application program, a session is established with the controlling application program. For a brief introduction to automatic logons, see “Automatic Logons” on page 158. For more information on automatic logon with independent LUs, see “Automatic Logons” on page 323.

- The name interpretation table function is not supported for independent LU sessions.
- The SSCP-PU session is optional and can be requested by the type 2.1 node during the XID exchange. However, the SSCP-PU session is one of the mechanisms that the type 2.1 node uses to transport network management device information (alerts) to the SSCP for delivery to the NetView program. An alternative mechanism used to report alerts is through a network management focal point function.

Attaching Peripheral Nodes to VTAM

This section describes attaching peripheral nodes to VTAM using:

- Channel connections
- External communication adapter connections.

Channel Connections

This section describes the following channel connections that can be used to attach a peripheral node to VTAM:

- Local non-SNA
- Local SNA
- Loop-adapter-attached.

Local Non-SNA Connection

Each local non-SNA major node defines a set of channel-attached (local) non-SNA terminals. Each minor node represents a non-SNA terminal (such as a 3277).

Local non-SNA devices are attached to an ES/9000 series processor through the Workstation Subsystem Controller.

You should always deactivate a local non-SNA major node before you take its terminals off-line.

To define a local non-SNA peripheral node:

1. Code an LBUILD definition statement followed by a collection of VTAM definition statements defining each channel-attached non-SNA terminal as part of a logical set (group) of channel-attached non-SNA terminals.

Remember the following:

- You do not need to code a definition statement for the non-SNA cluster controller (3272 or compatible device) to which the terminal is attached. Non-SNA terminals connected to a single cluster controller do not have to be defined in a single major node. Different terminals on the same controller can be defined to VTAM in different local non-SNA major nodes.
- You cannot specify the channel device name when you are activating the logical unit for a channel-attached non-SNA device; the channel device name must be specified on the LOCAL definition statement for the device.
- Code one LBUILD definition statement for each logical group (major node) of channel-attached non-SNA terminals.
- Do not code a PU definition statement. The physical unit for a channel-attached non-SNA device is VTAM's host physical unit (ISTPUS), which is

always active while VTAM is running. Therefore, only the local non-SNA major node and its logical units (representing the channel-attached devices) need to be defined and activated.

2. Code a LOCAL definition statement for each terminal (minor node) in the group. One or more LOCAL definition statements can be grouped with an LBUILD definition statement.

Following is an example of definition statements for a local non-SNA major node:

```
LC3270NS LBUILD          LOCAL NON-SNA ATTACHMENT
LOC3277  LOCAL  CUADDR=00A,  CHANNEL UNIT ADDRESS
                      TERM=3277,  TERMINAL TYPE

:
                      DLOGMOD=S3270,  DEFAULT LOGON MODE TABLE ENTRY
                      LOGAPPL=A50ACCTS  AUTOMATIC LOGON APPLICATION
```

Note: A local non-SNA terminal should not be defined to and activated by VTAM if its channel unit address is defined as an MVS console and allocated to console services. Activating a local non-SNA terminal whose channel unit address is in use by console services can cause VTAM, console services, or both to abend.

Local SNA Connection

Each local SNA major node defines a set of channel-attached (local) SNA cluster controllers. Each minor node represents an individual physical or logical unit.

MVS Channel-attached SNA devices can be either statically or dynamically defined. For an overview of defining local SNA devices dynamically, see “Dynamic Configuration of Channel-Attached Devices ” on page 147. For details, see “Dynamic Configuration of Channel-Attached Devices ” on page 287.

Local SNA devices must be attached to an ES/9000 series processor through a channel-attached block multiplexor channel.

Defining a Local SNA Connection: To statically define a local SNA device:

1. Define a local SNA major node by coding a VBUILD definition statement for the major node and a separate PU or LU definition statement for each minor node.
2. Code a PU definition statement for each physical unit (such as a cluster controller) in the major node.

Notes:

- a. A physical unit and all its logical units must be defined within a single major node.
- b. When activating a physical unit in a local SNA major node, supply the channel device name of the physical unit unless you have defined it in the PU definition statement. You can also specify the channel device name to override the value in the PU definition. To supply the channel device name, use the U operand of the VARY ACT command when activating the physical unit.
3. Code an LU definition statement for each logical unit placed after the associated PU definition statement. When you define a logical unit that logs on to VM, its name cannot be the same as any VM user ID in the system.

Sample Local SNA Connection: Figure 56 on page 230 is a sample configuration with local SNA devices:

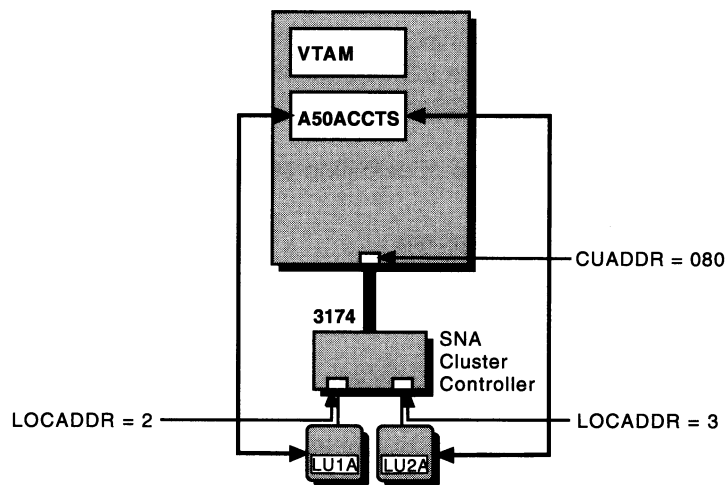


Figure 56. Local SNA Devices

Following is an example of definition statements for Figure 56:

LOCALSNA	VBUILD	TYPE=LOCAL	LOCAL SNA ATTACHMENT
LPU1	PU	CUADDR=080,	CHANNEL UNIT ADDRESS
:			
		MODETAB=LOG3174	LOGON MODE TABLE
*			
LU1A	LU	LOCADDR=2,	LOGICAL UNIT ADDRESS
:			
		MDLTAB=MDLTAB1,	MODEL NAME TABLE
		MDLENT=ENTRY3,	MODEL NAME TABLE ENTRY
		LOGAPPL=A50ACCTS	AUTOMATIC LOGON APPLICATION
*			
LU2A	LU	LOCADDR=3,	
:			
		MDLTAB=MDLTAB1,	MODEL NAME TABLE
		MDLENT=ENTRY3,	MODEL NAME TABLE ENTRY
		LOGAPPL=A50ACCTS	AUTOMATIC LOGON APPLICATION

MVS If VTAM requires operator assistance to create a connection to a physical unit in a local SNA major node, VTAM displays a message indicating that a connection request was denied because the physical unit is off-line. The connection request remains pending and the operator should either allow the connection request to complete by making the device available, or disallow the connection request by entering a VARY INOP command for the physical unit.

Enterprise Systems Connection (ESCON) Channel Attachment (ESA systems only): The Enterprise Systems Connection (ESCON) channel is a high-bandwidth host attachment facility that can be used to connect SNA and non-SNA 3174 controllers, 3172 interconnect controllers, 3746-900 controllers, and channel-attached hosts to VTAM running on VM/ESA, MVS/ESA, or VSE/ESA.

Loop-Adapter-Attached Connection

Loop-adapter-attached devices are defined in a local SNA major node, and they communicate with VTAM application programs as if they were channel-attached SNA physical units and logical units.

VM, VSE PU and LU definition statements can be used to define devices that are attached to the host processor through a loop adapter. SNA devices, such as the 8775 display terminal and the 3276 cluster controller, are defined as if they were channel-attached physical units; there is a PU definition statement representing the controller and an LU definition statement for each associated logical unit. Non-SNA devices are defined as logical units associated with a single PU definition statement that represents physical unit functions provided by the loop adapter.

External Communication Adapter (XCA) Connections

This section describes the local area networks that can connect to VTAM through an XCA, the 3172 interconnect controller.

VTAM and the 3172 interconnect controller support the following types of LANs:

- Token-ring
- Fiber Distributed Data Interface (FDDI)
- Carrier sense multiple access with collision detection (CSMA/CD) 802.3.

This support allows SNA applications to access the supported LANs. TCP/IP and SNA applications can use the same physical attachment to a 3172 interconnect controller at the same time.

Note that VTAM connected through a 3172 interconnect controller cannot own an NCP across a local area network.

Defining an XCA Configuration

To define a 3172 interconnect controller configuration, code the following:

- One external communication adapter major node to represent the physical unit in the 3172 interconnect controller. This is not required, but is used for network management purposes. If you are running the NetView program, it is recommended that you code this definition statement. By defining this physical unit, you can have the 3172 interconnect controller forward the same type of information that a 3174 cluster controller can forward. For example, any alerts detected by the 3172 interconnect controller can be forwarded.
- One external communication adapter major node for each LAN connected to the 3172 interconnect controller.

Note: If you have both subarea nodes (type 4 and type 5 nodes) and peripheral nodes (type 1, type 2, type 2.1, and subarea nodes that appear as type 2.1 nodes) attached to the LAN, you must code two GROUP definition statements in this external communication adapter major node (one for the peripheral devices, and one for the subareas).

- Major nodes for any peripheral devices connected to the LAN.

Example XCA Configuration

For example, following is a 3172 interconnect controller configuration:

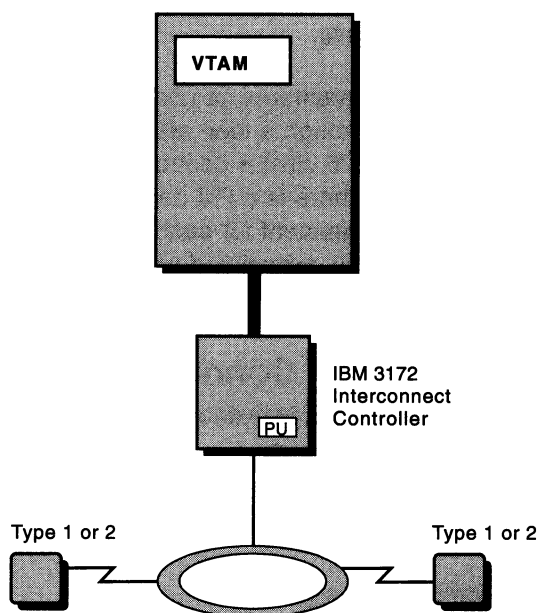


Figure 57. XCA Connection in a Single-Domain Environment

For the configuration in Figure 57, code the following:

1. An external communication adapter major node to represent the physical unit in the 3172 interconnect controller:

XCACON1	VBUILD TYPE=XCA	XCA MAJOR NODE
PORT1	PORT MEDIUM=BOXMGR,	3172 MANAGER
	CUADDR=500	CHANNEL UNIT ADDRESS
*		
GROUP1	GROUP ISTATUS=ACTIVE	ACTIVATED AT GEN
LINE1	LINE	
PU1	PU	

2. Another external communication adapter major node for the token ring:

XCACON2	VBUILD TYPE=XCA	XCA MAJOR NODE
PORT2	PORT MEDIUM=RING,	TOKEN-RING
	SAPADDR=4,	SERVICE ACCESS POINT ADDRESS
	ADAPNO=1,	ADAPTER NUMBER
	CUADDR=BC0	CHANNEL UNIT ADDRESS
*		
GROUP2A	GROUP DIAL=YES,	SWITCHED PERIPHERAL NODE
	CALL=INOUT,	CAN DIAL IN OR OUT
	ISTATUS=ACTIVE	ACTIVATED AT GEN
LINE2A	LINE ANSWER=ON	PU CAN DIAL IN
PU2A	PU	
GROUP2B	GROUP DIAL=YES,	SWITCHED PERIPHERAL NODE
	CALL=INOUT,	CAN DIAL IN OR OUT
	ISTATUS=ACTIVE	ACTIVATED AT GEN
LINE2B	LINE ANSWER=ON	PU CAN DIAL IN
PU2B	PU	

3. A switched major node for the two peripheral nodes attached to the LAN:

```

SWNODE1  VBUILD TYPE=SWNET,
          MAXGRP=2,          MAXIMUM NUMBER OF UNIQUE GRPNM
          MAXNO=2            MAXIMUM NUMBER OF UNIQUE DIALNO
          PUA      PU      ADDR=C1,
          DYNLU=NO,          NO DYNAMIC ALLOCATION OF CDRSCS
          PUTYPE=2,          PHYSICAL UNIT TYPE

          :

          IDBLK=012,        BLOCK IDENTIFICATION
          IDNUM=00012        IDENTIFICATION NUMBER

*
LANPATHA PATH GRPNM=GROUP2A, DIAL-OUT PATH
              DIALNO=0104400000000013, PLACE HOLDER, SAP, AND MAC
              GID=2             PATH GROUP
LUA          LU      LOCADDR=1    LOGICAL UNIT ADDRESS
*
PUB          PU      DYNLU=NO,    NO DYNAMIC ALLOCATION OF CDRSCS
              PUTYPE=2,          PHYSICAL UNIT TYPE

          :

          CPNAME=NAME2        CPNAME IDENTIFICATION: TYPE 2.1 NODE

*
LANPATHB PATH GRPNM=GROUP2B,    DIAL-OUT PATH
              DIALNO=0104400000000014, PLACE HOLDER, SAP, AND MAC
              GID=2             PATH GROUP
LUB          LU      LOCADDR=2    DEPENDENT LOGICAL UNIT ADDRESS
*

          :

```

Note: Switched PUs and LUs can be defined to VTAM using the switched or model major node. See “Switched SDLC Connection” on page 235, “Defining Switched Resources Dynamically” on page 143 and “Dynamically Defining Switched Resources” on page 260.

Automatically Generating Lines and Physical Units

You can enable VTAM to automatically generate lines and physical units on an external communication adapter major node. To use this function, code the AUTOGEN operand on the GROUP definition statement in any external communication adapter major node where you have coded DIAL=YES. VTAM uses the values that you specify on the AUTOGEN operand to build the definitions for lines and physical units.

Names for the lines and physical units are generated using the naming convention 12345678 where:

- 1 is the character you coded on the AUTOGEN operand. The first character is for lines, and the second character is for physical units. If you do not code a character on the AUTOGEN operand, VTAM uses the IBM defaults: O for lines and Q for physical units.
- 2 – 5 represent the channel unit address specified in the CUADDR operand on the PORT definition statement.
- 6 – 8 represent a sequential number in the range of X'000'–X'FFF'.

For example, if you want to enable VTAM to generate 15 lines and physical units where each LINE name begins with L and each PU name begins with P, code the following:

```
PORT2    PORT  MEDIUM=RING,          TOKEN-RING
          SAPADDR=4,                SERVICE ACCESS POINT ADDRESS
          ADAPNO=1,                  ADAPTER NUMBER
          CUADDR=BC0                 CHANNEL UNIT ADDRESS

*
GROUP2A  GROUP DIAL=YES              SWITCHED PERIPHERAL NODE
          CALL=INOUT                 CAN DIAL IN OR OUT
          ISTATUS=ACTIVE,            ACTIVATED AT GEN
          AUTOGEN=(15,L,P)           GENERATE LINES AND PUS
```

This generates 15 lines and 15 physical units. The lines will be named in the range L0BC0000–L0BC000F. The physical units will be named in the range P0BC0000–P0BC000F.

Integrated Communication Adapter (ICA) Connections (VM, VSE)

You can use the following connections to attach peripheral nodes through an ICA in the ES/9000 series processors:

- Nonswitched SDLC lines
- Switched SDLC lines
- **VSE** X.21 leased and switched lines
- Token-ring networks
- X.25 permanent and virtual switched circuits
- BSC.

You can also attach a peripheral node to an ES/9000 series processor using channels. Refer to “Channel Connections” on page 228 for information about this.

You can use all of the above connections with an ES/9000 series processor except BSC.

Start-stop support is defined to VTAM as an SNA physical and logical unit. The start-stop protocol must be converted or enveloped in an SNA data stream using a 3708 protocol converter.

Nonswitched SDLC Connection

Use the channel-attachment (CA) major node to define a nonswitched SDLC connection.

GROUP definition statement

You can define some of the information for the LINE, PU, and LU definition statements using the GROUP definition statement. The operands will then sift down to the other definition statements unless you override them in the other definition statements.

LINE definition statement

If a line is defined in more than one line group, only one of the groups can be active at a time.

PU definition statement

For each physical unit attached to the link, you code a PU definition statement. VTAM uses the PU definition statement to define the link

station for an adjacent host processor (physical unit type 5) or communication controller (physical unit type 4).

LU definition statement

You must have a separate LU definition statement for each logical unit associated with a peripheral node on a nonswitched line.

The following is a sample definition statement for an SDLC 3274 cluster controller connected through an ICA:

```
CAA81    VBUILD TYPE=CA
GA81     GROUP LNCTL=SDLC,          SDLC LINE CONTROL

:
:                                     DIAL=NO          NONSWITCHED LINK
*
LA81     LINE ADDRESS=A81,          COMMUNICATIONS ADAPTER LINE ADDRESS

:
:                                     ISTATUS=ACTIVE    VTAM INITIAL STATUS
*
PA8101   PU   ADDR=C1,              PHYSICAL UNIT ADDRESS
          PUTYPE=2,                  PHYSICAL UNIT TYPE
          MAXOUT=7,                  NUMBER OF FRAMES FOR LINK LEVEL RESPONSE
          MAXDATA=265,               MAXIMUM DATA SIZE

:
:                                     ISTATUS=ACTIVE    VTAM INITIAL STATUS
*
LUA8101  LU   LOCADDR=2,            LOGICAL UNIT ADDRESS

:
:                                     ISTATUS=ACTIVE    VTAM INITIAL STATUS
```

Switched SDLC Connection

To define a switched connection to VTAM through an ICA, you must define the following:

- A switched major node to define the switched line and physical unit resources attached to the communication adapter.
- A channel-attachment major node to define the resources required to establish a temporary connection in VTAM; the switched line and physical unit resources are defined in the channel-attachment major node.

The association between a physical resource and its logical definition in VTAM is made when the switched connection is established.

You can dynamically define switched connections. See “Defining Switched Resources Dynamically” on page 143.

Definition Statements for Switched SDLC: The switched major node contains the definition of the physical and logical units.

Use the following definition statements when coding the switched major node:

LINE definition statement

If a line is defined in more than one line group, only one of the groups can be active.

PU definition statement

The PU and LU definition statements used in a switched major node are very similar to the PU and LU definition statements used in a local SNA major node. The PU definition statement defines a peripheral or subarea physical unit connected by a switched SDLC line.

LU definition statement

The LU definition statement must follow the PU definition statement that defines the peripheral node with which the logical unit is associated. The LU definition statement is not applicable for a switched subarea connection.

Sample Switched Major Node: Following is a sample switched major node.

```

SWSNA032 VBUILD TYPE=SWNET,          SWITCHED MAJOR NODE

:
      MAXNO=6          MAXIMUM UNIQUE TELEPHONE NUMBERS
*
P032A  PU  ADDR=C1,          SDLC ADDRESS OF STATION
      ANS=CONTINUE,        AUTOMATIC NETWORK SHUTDOWN - CONTINUE
      ISTATUS=ACTIVE,      INITIAL STATUS OF ACTIVE
      MAXDATA=256,         MAXIMUM DATA TRANSFER TO PERIPHERAL NODE
      MAXOUT=7,            MAXIMUM FRAMES FOR LINK LEVEL RESPONSE
      MODETAB=MODETAB1,    LOGON MODE TABLE
      PACING=7,            PACING VALUE

:
      PUTYPE=2          SNA CLUSTER CONTROLLER
*
PATH01  PATH  DIALNO=1234567,    TELEPHONE NUMBER
      GID=1,            SWITCHED LINE GROUP IDENTIFIER
      PID=1,            DIAL PATH IDENTIFIER

:
      REDIAL=3          NUMBER OF DIAL RETRIES
*
LU03201  LU  LOCADDR=2,        DEPENDENT LOGICAL UNIT ADDRESS
      ISTATUS=INACTIVE,    INITIAL STATUS INACTIVE

:
      LOGAPPL=A50ACCTS    CONTROLLING PRIMARY LU
  
```

You can define multiple sets of switched devices. This allows the VTAM operator to use the VARY command to activate a subset of all the switched devices.

Controlling Dial-In Requests: If you want a physical unit to dial to VTAM on a switched line, you must code the PU's station identification number so that VTAM can identify the physical unit. The station identifier is a 48-bit number that is used in the XID exchange during the dial procedure. This station identifier must be unique for each station in the network.

If the peripheral node is a type 2.1 physical unit, you can use the CPNAME operand on the PU definition statement to specify the station identification number. If the peripheral node is a VTAM host defined as a type 2.1 physical unit (see “Type 2.1 Casual Connection” on page 407), you must use the CPNAME operand; and the value of the CPNAME operand must be the same as the value of the SSCPNAME start option in the attaching VTAM type 2.1 peripheral node. The IDBLK and IDNUM operands are ignored.

If the peripheral node is a type 1 or 2 physical unit, then you must use the IDBLK and IDNUM operands on the PU definition statement. VTAM uses the IDBLK and IDNUM values to build the station identifier.

The value that you code for IDBLK is a three digit hexadecimal number. Each device type has its own device addressing scheme; you can find the IDBLK value in the component description manuals for the device. Note that this only applies to physical units; you do not code IDBLK for logical units.

Because a subarea physical unit can have multiple connections, a connection identifier (IDNUM) is used to distinguish the multiple PU definition statements that may apply to a subarea physical unit.

As soon as a dial-in line has been activated and placed in answer mode (during VTAM startup or by the VTAM operator), the physical unit can dial that line's number and establish communication with VTAM. A physical unit can be defined as call-in only, call-out only, or both.

Controlling Dial-Out Requests, ID Exchange, and Controlling Disconnection:

For information on controlling dial-out requests, ID exchange, and controlling disconnection, see “Controlling Dial-Out Requests” on page 254, “ID Exchange” on page 254, and “Controlling Disconnection” on page 255.

X.21 Switched Line (VSE)

The X.21 line protocol, when coded in a switched line definition statement, only applies to type 2 PUs.

X.21 Short Hold Mode/Multiple Port Sharing (SHM/MPS): SHM/MPS is a switched line protocol that VTAM can use to provide support for many PUs over a small number of lines. To share these lines, switched calls are dropped during inactive periods and are reestablished when data traffic starts.

The SHOLD operand on the GROUP definition statement tells VTAM whether the group is to operate under X.21 SHM/MPS. Using SHOLD, you must specify these values:

free

The free time, in seconds, after the dial is complete.

npoll

the number of nonproductive polls to allow before a connection is dropped.

Be careful not to set the preceding values too low; otherwise, the connections drop too quickly. This can cause VTAM to operate inefficiently, with frequent reconnections and network disruptions.

For NPSI connections, X.21 SHM/MPS is defined on the PATH definition statement using the SHOLD, SHM, and SHMTIM operands. See “Switched Major Node” in the *VTAM Resource Definition Reference* for an explanation of these operands.

X.21 Call Progress Signal Retry: When an X.21 call fails, the ICA tells VTAM whether the error is due to a short term condition (such as a brief moment of line noise) or a long term condition (such as a busy signal). The condition is described as either *retriable* or *non-retriable*, respectively.

With X.21 call progress signal (CPS) retry, you may opt to have VTAM retry *non-retriable* calls. You control the way VTAM retries these calls with the CPS retry table, which you load with the CPSTAB start option or the MODIFY TABLE command.

In the CPS retry table, you specify:

- CPS number
- Number of times VTAM may retry a given CPS
- Amount of time VTAM must wait between each retry attempt.

While VTAM is waiting to retry a given CPS, the line is used for other calls.

You may decide to use the CPS retry table to retry *retriable* calls as well as non-retriable calls. There are a few reasons for doing this. First, the CPS retry table allows VTAM to use the line for other calls during the delay. Second, the CPS retry table allows you to code different retry limits and retry delays for each signal. To retry *retriable* calls with the CPS retry table, you must set the RETRYTO operand of the LINE statement to zero. A sample default call process signal table follows.

```
      EJECT
      PUNCH    ' PHASE ISTCPS10,S'
ISTCPS10 CPSTAB  START
          CPSENTRY CPSCODE=41,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=42,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=43,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=44,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=45,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=46,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=47,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=48,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=49,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=51,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=52,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=71,CPSRCNT=5,CPSRDLY=5
          CPSENTRY CPSCODE=72,CPSRCNT=5,CPSRDLY=5
          CPSTAB  STOP
      END
```

X.21 Direct Call: If an X.21 direct call line is available, its PATH definition statement should immediately follow the switched physical unit's PU definition statement. This ensures that VTAM attempts to call the physical unit on the direct call line before trying a dial line.

Token-Ring Connection through the ICA

To define a token-ring connection and physical unit resources, do the following:

- Code a LAN major node to define local area network physical resources.
- Code a switched major node to define the logical connections over the token ring.

Coding the LAN Major Node: One VBUILD definition statement is needed for each local area network, but it can contain different link groups. Each link group has a unique GROUP definition statement.

Sample LAN Major Node: Following is a sample local area network major node:

LAN1	VBUILD	TYPE=LAN	LAN MAJOR NODE
PORT1	PORT	CUADDR=840,	LAN CHANNEL UNIT ADDRESS
		LANCON=(10.1,3),	LAN CONNECTION VALUES
		MACADDR=00000031,	LAN MEDIUM ACCESS CONTROL ADDRESS
		MAXDATA=2012,	LAN MAXIMUM DATA LENGTH
		MAXSTN=8,	LAN MAXIMUM STATIONS
		:	
		SAPADDR=4	LAN SERVICE ACCESS POINT ADDRESS
GP1LAN	GROUP	DIAL=YES,	SWITCHED ATTACHMENT
		ISTATUS=INACTIVE,	
		LNCTL=SDLC	APPEAR TO VTAM AS SDLC LINES
LN1001	LINE	CALL=INOUT	INCOMING/OUTGOING CALLS
PU1001	PU		
LN1002	LINE	CALL=INOUT	INCOMING/OUTGOING CALLS
PU1002	PU		
		*	

Coding the PORT Definition Statement: Code a PORT definition statement to identify the address of the access protocol group in the token-ring subsystem of the ES/9000 series processor. Its keywords define parameters that are passed to the subsystem.

Figure 58 on page 240 illustrates how these parameters are used by the subsystem.

The values of the MACADDR, SAPADDR, and CUADDR operands are used to route information between the LAN and VTAM. When information flows from VTAM down to the LAN, the first digit of the CUADDR operand defines the channel used to route the information to the subsystem.

When information flows from the LAN up to VTAM, the MACADDR operand is used to route the information to the subsystem. The subsystem builds a table from the SAPADDR and CUADDR operands and uses the table to route the information to the channel unit address (CUA) corresponding to the service access point (SAP).

Coding the GROUP Definition Statement: The GROUP definition statement defines a line group for type 2 PUs attached to the token-ring network.

Coding the PORT definition statement causes the GROUP definition statements that follow to use the default of LNCTL=SDLC.

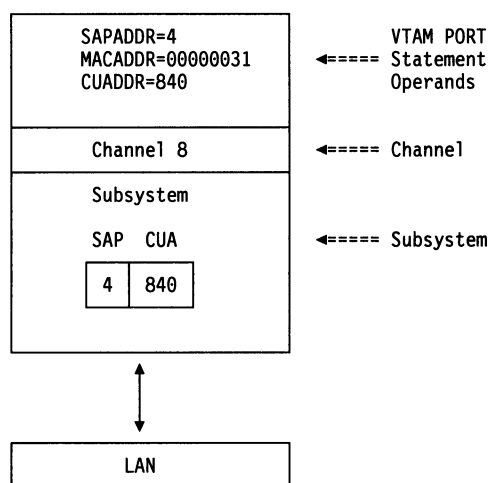


Figure 58. IBM Token-Ring Subsystem Controller and PORT Definition Statement Operands

The GROUP name is not used for incoming calls, only outgoing calls. For incoming calls, all the lines reside in the same pool and can be used for any incoming call. Therefore, you should code only one GROUP definition statement.

VTAM resolves outgoing calls by searching for a line starting at the top of the pool and searching down; for incoming calls VTAM searches for a line starting at the bottom of the pool and searching up. It is easy for a conflict to occur if you code different CALL options for one group of lines. Therefore, you should specify the same CALL option for lines under the same GROUP definition statement.

Automatically Generating Lines and Physical Units: You do not have to explicitly code LINE and PU definition statements for a LAN major node that has DIAL=YES coded on the GROUP definition statement. Instead, you can enable VTAM to automatically generate lines and physical units. For information on using this function, see “Automatically Generating Lines and Physical Units” on page 233.

Coding the Channel Unit Address: The CUADDR operand, which is required, defines the channel unit addresses used for the attached LAN. The address that is specified in CUADDR is a 3-digit hexadecimal channel address. The first digit of the channel unit address is the channel number and is in the range hex 0–F. The second and third digits are the subchannel address, in the range hex 00–FB.

VTAM uses the channel unit address and the next three sequential addresses as channel unit addresses. For example, if you specify CUADDR=848, VTAM assigns 848, 849, 84A, and 84B as the channel unit addresses. When VTAM attaches to the LAN, it verifies that the device at each of the four channel unit addresses is defined to the operating system as a **VM** 3088 device, **VM** ICA device type and industrial LAN adapter, or a **VSE** 3705,10 device.

VM Following is a sample HCPRIO (real device addresses for VM/ESA) for an ES/9221 LAN.

```
RDEVICE DEVTPE=ICA,ADAPTER=ILAN,DEVNO=(B00,12)
RDEVICE DEVTPE=ICA,ADAPTER=ILAN,DEVNO=(B20,12)
RDEVICE DEVTPE=ICA,ADAPTER=ILAN,DEVNO=(B40,12)
RDEVICE DEVTPE=ICA,ADAPTER=ILAN,DEVNO=(B60,12)
```

VSE At IPL time, you must supply an ADD statement for each of the channel unit addresses. You must code the device type on the ADD statements as 3705,10, as shown in the following example:

```
ADD 840,3705,10
ADD 841,3705,10
ADD 842,3705,10
ADD 843,3705,10
```

VSE You must also include the channel unit addresses in the number of I/O devices on the IOTAB (I/O table) macroinstruction. IOTAB defines the storage to be set aside for various system device tables.

Channel unit addresses in these examples are defined to support up to three access protocol groups, referred to as a continuously executing transfer interface (CETI) group. Using CETI reduces the number of START I/O commands and interrupts required to transfer data. For example, VTAM can use addresses 840, 841, 842, and 843 for the access protocol group; a second CETI group can use addresses 844, 845, 846, and 847; and a third CETI group can use addresses 848, 849, 84A, and 84B.

Coding the Switched Major Node: A station attached to the token-ring must be defined to VTAM using a switched major node.

The PU and LU definition statements define peripheral physical and logical units on switched lines that can be dialed in to or dialed out from the ES/9000 series processor. These lines are defined in switched line groups in the LAN major node. For dial-out operations, the PATH definition statement defines the possible paths to be used to establish connection between the ES/9000 series processor and the physical unit.

Following is a sample switched major node for LAN connections:

```
TRNSW840 VBUILD TYPE=SWNET,

:
MAXNO=50                                MAXIMUM UNIQUE DIALNO
*
PU2      PU      IDBLK=017,              BLOCK IDENTIFICATION
                                IDENTIFICATION NUMBER
                                LAN ACKNOWLEDGMENT VALUES
                                LAN CONNECTION VALUES
                                LAN INACTIVITY VALUE
                                LAN RESPONSE VALUES
                                MAXPATH=4,

:
MACADDR=400000000013                  LAN MEDIUM ACCESS CONTROL ADDRESS
*
LANPATH2 PATH  GRPNM=GP2LAN            LAN GROUP IN LAN MAJOR NODE
DIALNO=400000000013
LU2      LU      LOCADDR=2,

:
VPACING=0                              NO VTAM TO BOUNDARY FUNCTION PACING
*
PU3      PU      IDBLK=017,              BLOCK IDENTIFICATION
```

```

                                IDNUM=00017,          IDENTIFICATION NUMBER
:
                                LANSW=YES             LAN STATION IS LAN CAPABLE
*
LU3      LU      LOCADDR=2          DEPENDENT LOGICAL UNIT

```

If you have a switched major node defined in your network, you can use a copy of that definition for the switched major node in the LAN environment, and add the operands required for the LAN environment. Operands coded in your present definition that are not required for the LAN environment are allowed but have no meaning.

Multiple Subsystems: There can be multiple token-ring subsystems in an ES/9000 series processor. Each subsystem is attached to a different channel and supports up to three connections to a LAN. If more than one connection is made through a subsystem or if more than one subsystem is installed, you must be aware of certain considerations when coding the operands on the PORT definition statement.

Figure 59 illustrates three subsystems installed in the ES/9000 series processor. In the figure, all of the users connected to the LANs through the subsystems are VTAMs. However, application programs other than VTAM can be users of the subsystems.

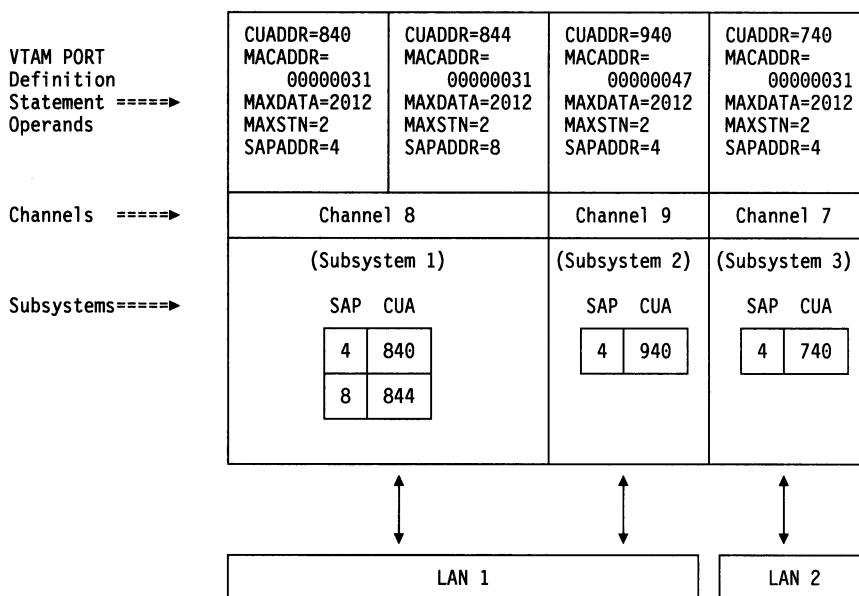


Figure 59. Multiple Subsystems in ES/9000 Series Processor

The ES/9000 series processor can maintain multiple connections to the same token-ring subsystem. Subsystem 1 has two connections between VTAM and LAN 1. When a connection is made to LAN 1, VTAM passes parameters coded on operands of the PORT definition statement to the subsystem.

The values coded on the MACADDR, MAXDATA, and MAXSTN operands of the first VTAM to connect to Subsystem 1 set parameters used for all connections to the subsystem. Therefore, subsequent connections to Subsystem 1 must use the same values for these operands. The value for MAXSTN must include the total

number of stations for all connections to Subsystem 1. The values coded for the CUADDR and SAPADDR operands must be different for each VTAM connected to Subsystem 1.

The ES/9000 series processors can also support multiple connections to the same LAN. Subsystem 1 and Subsystem 2 are both connected to LAN 1. When information flows from the LAN up to VTAM, the MACADDR operand identifies the subsystem to which the information should be routed. The values of the MACADDR operand must be different for each subsystem connected to the same LAN. Subsystem 3 has the same value for MACADDR as Subsystem 1, but the subsystems are connected to different LANs.

Token-Ring Security: You can restrict users that can dial in to your VTAM. The name that you code for the GROUP definition statement in the LAN major node must match the GRPNM operand on the PATH definition statement in the switched major node. Therefore, the value that you code in the GRPNM operand on the PATH definition statement in the switched major node specifies the users in a group that can dial your system. Only the users in the group that you specify on the LAN major node can access your system.

X.25 Connection

You define an X.25 packet switched data network (PSDN) using a packet major node and a switched major node for any switched lines.

Note: You do not have to use an X.25 PSDN between two DTEs. See "X.25 DTE-to-DTE Support through the ICA" on page 247.

Defining an X.25 Configuration: To define an X.25 configuration, do the following in the packet major node:

1. Code a PORT definition statement to identify the X.25 link-level characteristics, the PSDN network type, and the X.25 packet-level characteristics. Because both nonstandard default window and packet sizes have been subscribed to at location A in Figure 60 on page 245, the PLENGTH and PWINDOW operands are coded in the sample. The CUADDR operand defines the channel unit address pair that is used for X.25 protocols. The lower address of the pair must be specified first, and it represents a physical address on an ES/9000 series processor.

For X.25 support, you must define the address pair on the PORT definition statement to the operating system.

- **VSE** At IPL of VSE, you must supply an ADD statement for each of the channel unit addresses of your X.25 ports. You must code the device type on the ADD statements as 3705,10, as shown following:

```
ADD 800,3705,10
ADD 808,3705,10
```

You must also include the X.25 ports in the number of I/O devices on the IOTAB (I/O table) macroinstruction. IOTAB defines the storage to be set aside for various system device tables.

- **VM** Following is a sample HCPRIO (real device addresses for VM/ESA) for the address pair on the PORT definition statement:

```
RDEVICE DEVTYPE=ICA,ADAPTER=SDLC,DEVNO=(B00,12)
RDEVICE DEVTYPE=ICA,ADAPTER=SDLC,DEVNO=(B20,12)
```

2. Code a VCPARMS definition statement to identify the flow control parameter characteristics of a group of virtual circuits. Although this definition statement is optional, you code it to identify the use of nonstandard default packet and window sizes for the PVC.
3. Code a GROUP definition statement. VTAM treats X.25 virtual circuits as if they were SDLC lines. The DIAL operand on the GROUP definition statement determines whether the lines in that group are nonswitched (PVCs) or switched (SVCs). For switched lines, you must also code a switched major node.
4. Code a LINE definition statement for each virtual channel allocated for your use by the PSDN. When a LINE definition statement appears under a nonswitched group, the ADDRESS operand specifies the actual logical channel identifier (LCID) assigned by the X.25 PSDN. For switched lines, the LCID is chosen from the range set on the VCALLS operand of the PORT definition statement. Operands on the LINE definition statement can be coded to define the name for the line and certain procedural options to be used for the line.
5. Code a PU definition statement for PU001 to represent the remote device that is connected over an X.25 PVC. Also, code a PU definition statement under a switched line for an X.25 SVC connection. Then you must also code PU and LU definition statements in a switched major node for the SVC connection.

Note: VTAM uses the PU definition statement to define the link station for an adjacent host processor (physical unit type 5) or communication controller (physical unit type 4).

6. Code an LU definition statement to identify the remote device that is connected over an X.25 PVC.

This definition statement is not required for SVC connections. You code the LU definition statement for the remote device of an SVC in the switched major node. Each logical unit associated with a peripheral node on a nonswitched line must be represented by a separate LU definition statement.

Sample X.25 Based Configuration: Figure 60 on page 245 shows the sample X.25 based configuration that is discussed in this section. In the figure, all physical access lines to the X.25 network are represented by solid lines, and the virtual circuits are represented by dotted lines. It is assumed that the access line speed at all X.25 ports is sufficient to accommodate traffic. X.25 subscription options and connectivity from locations X and Y to location A have been planned as follows:

Location A—ES/9000 series processor

Two PVCs and one SVC (two-way call) have been subscribed to. The PVC is used for connecting to the 3274 Control Unit at location X, and the SVC is used for connecting to the 3174 control unit at location Y. Reverse charging acceptance has been requested, allowing charges to accumulate at the host end of the virtual call to location Y. Non-standard default packet size of 256 and window size of 7 have been requested for all circuits on the port.

Location X—IBM 3274

A PVC has been subscribed to and is used to establish connectivity to an ES/9000 series processor at location A.

Location Y—IBM 3174

An SVC (two-way call) has been subscribed to and is used to establish connectivity to location A. Reverse charging has been subscribed to, allowing charges to accumulate at the host end of the virtual call.

The IBM 3274 at location X and the IBM 3174 at location Y must be customized to reflect link-level characteristics for successful attachment to the PSDN.

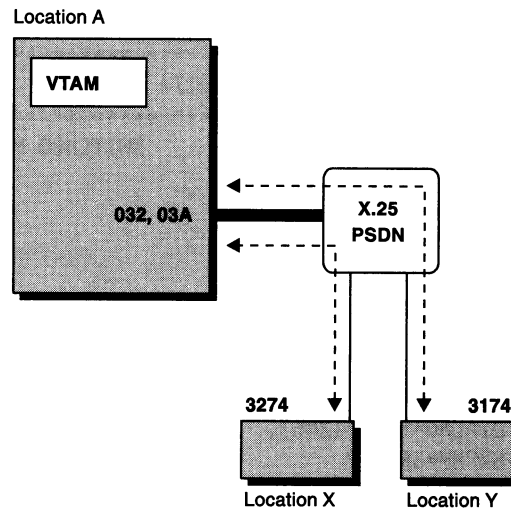


Figure 60. X.25 Based Configuration in Single-Domain Environment

All X.25 circuits subscribed to at Location A must appear under one packet major node definition. Remote SNA devices are defined in the packet major node for connectivity established with a PVC. A switched major node is required for definition of remote devices that establish connectivity using SVCs.

Following is a sample packet major node at Location A:

CALOCA25	VBUILD	TYPE=PACKET	X.25 PORT
	PORT	CUADDR=(800,808),	X.25 CHANNELS
		NETTYPE=3,	
		VCALLS=(,002,002,,),	SVC RANGE, INOUT CIRCUITS
		PLENGTH=256,	SVC PKT SIZES (NONSTD)
		PWINDOW=7,	SVC WINDOW SIZE (NONSTD)
		CHARGACC=YES	REVERSE CHARGING ACCEPTED
	VCPARMS	LC=(1,2),	
		PLENGTH=256,	PVC PKT SIZE (NONSTD)
		PWINDOW=7	PVC WINDOW SIZE (NONSTD)
GROUP1	GROUP	LNCTL=SDLC,	PVC LINE GROUP
		DIAL=NO,	
		ISTATUS=INACTIVE	
LINE001	LINE	ADDRESS=001	USED FOR 3274 AT LOCATION X
PU001	PU	PUTYPE=2,	
		ADDR=C1	
LU32741	LU	LOCADDR=2,	
		DLOGMOD=D4C32782,	
		LOGAPPL=CICS	
LU32742	LU	LOCADDR=3,	
		DLOGMOD=D4C32782,	
		LOGAPPL=CICS	
GROUP2	GROUP	LNCTL=SDLC,	SVC LINE GROUP
		DIAL=YES,	
		ANSWER=ON	
LINE002	LINE		USED FOR 3174 AT LOCATION Y
:			

PU002 PU

Following is a sample switched major node at Location A in Figure 60 on page 245:

```
* ***** *
* SWLOCA01 - SWITCHED MAJOR NODE FOR 3174 CALL *
* ***** *
SWLOCA01 VBUILD TYPE=SWNET,          SWITCHED MAJOR NODE
              MAXGRP=3,
              MAXNO=3
SWPU003 PU    ADDR=C1,
              PUTYPE=2,
              MAXPATH=1,
              MAXDATA=262,
              IDBLK=017,
              IDNUM=70933
              PATH DIALNO=(47031205),    CALLED DTE ID
              GRPNM=GROUP2              REFER TO SVC GROUP NAME
SWLU003 LU    LOCADDR=2,
              DLOGMOD=D4C32762,
              LOGAPPL=CICS,
              PACING=1,
              VPACING=2
```

The PU and LU definition statements define physical units and logical units on the remote end of the SVC.

VTAM Common Network Services (VCNS) (VM): For each X.25 port used with VCNS, you must code a GROUP definition statement with DIAL=NO specified followed by one LINE definition statement with USER=VCNS specified. Do not code PU or LU definition statements. This collectively identifies all virtual circuits owned by VCNS and associated with an X.25 port.

The port cannot be used for VCNS-based communication until the line is activated.

Following is a sample line (P11AVCNS) owned by VCNS that would be coded in a packet major node:

```
GROUP1 GROUP LNCTL=SDLC,          PVC LINE GROUP
              DIAL=NO,
:
P11AVCNS LINE ISTATUS=INACTIVE,    LINE FOR VCNS VIRTUAL CIRCUITS
              USER=VCNS
```

X.25 Optional User Facilities: CCITT Recommendation X.25 defines a set of optional user facilities that, when implemented, provide extended flow control at the DTE/DCE interface and security and control options for SVCs. While some facilities require support by the X.25 DTE, others are provided solely by the PSDN common carrier.

The following table summarizes all X.25 optional user facilities supported by VTAM and identifies the operands and definition statements you must code to implement these facilities. The table also identifies the VTAM definition statements and operands required to implement these facilities. For a description of each optional user facility supported by VTAM for SNA circuits, refer to the *Planning for NetView, NCP, and VTAM*.

Table 15. VTAM Optional User Facility Definition

Optional User Facility	Operands and Definition Statements
Extended Packet Sequence Numbering	PMOD on PORT
Nonstandard Default Window Size	PWINDOW on PORT or on VCPARMS
Nonstandard Default Packet Size	PLENGTH on PORT or on VCPARMS
One-Way Logical Channels	VCALLS on PORT
Reverse Charging	CHARGE on PORT and DIALNO on PATH
Reverse Charging Acceptance	CHARGACC on PORT
Closed User Group	DIALNO on PATH
Recognized Private Operating Agency	DIALNO on PATH
Flow Control Parameter Negotiation	PWINDOW on PORT or on VCPARMS
Throughput Class Negotiation	NONE
Default Throughput Class Assignment	NONE
Incoming and Outgoing Calls Barred	NONE
Incoming and Outgoing Calls Barred with Closed User Group	NONE

X.25 DTE-to-DTE Support through the ICA

Two DTEs can communicate without an intervening X.25 packet switched data network (PSDN). Instead, you can use a modem or modem eliminator between the two DTEs. Rather than X.25's defining the interface between the two DTEs, one of the DTEs can assume the role of a data communication equipment (DCE).

To enable this support, use the NETTYPE operand on the PORT definition statement of the packet major node. NETTYPE specifies the type of network to which the device wants to connect. For example, if you code NETTYPE=DCE, that means the device wants to connect to a device that will act as a DCE. You must specify NETTYPE=DTE for VTAM to assume the DCE role.

If you use a DTE-to-DTE connection, one of the DTEs must be coded as NETTYPE=DCE. If both are coded as NETTYPE=DTE or as NETTYPE=DCE, you will get unpredictable results. VTAM can act as a DCE on any ES/9000 series Telecommunications Subsystem X.25 port attached through a leased line.

When you code NETTYPE=DTE, you must code the VCALLS operand on the PORT definition statement from the perspective of the remote DTE. This means that the incoming channels that you code will be used as outgoing channels by this DTE.

Figure 61 on page 248 shows a sample DTE-to-DTE configuration.

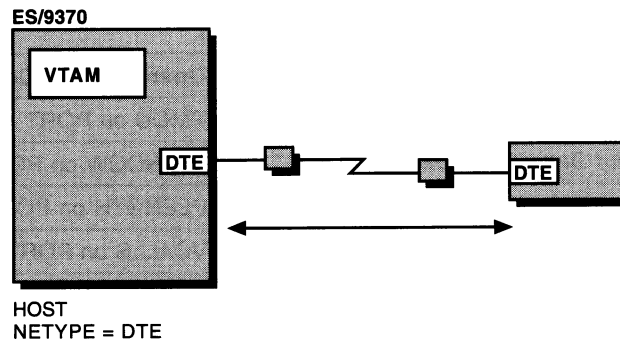


Figure 61. DTE-to-DTE Configuration in Single-Domain Environment

BSC Connection

To connect to a BSC device, you must code the following:

LINE definition statement

Defines the name, channel unit address, and procedural options for a BSC nonswitched line. If a line is defined in more than one line group, only one of the groups can be active.

CLUSTER definition statement

Defines the resource name, type, polling characteristics, and communication procedures for a BSC 3270 cluster controller.

TERMINAL definition statement

Defines the resource name, type, features, and communication procedures for a BSC terminal attached to a 3270 cluster controller. This statement also defines the device address (device selection character) by which VTAM contacts the terminal (multipoint line control).

For example, a printer attached to a 3275 is defined by coding `FEATUR2=PRINTR` on the `TERMINAL` definition statement for the 3275, not by coding a separate `TERMINAL` definition statement for the printer.

The following sample defines a BSC 3274 cluster controller connected to an ES/9000 series communication adapter. Operands normally associated with an SDLC attached peripheral node can also be used with a peripheral node that is attached using BSC (for example, the model terminal name table [MDLTAB operand] and the exact entry in the table to be used for the specific device [MDLENT operand]).

CAA83	VBUILD	TYPE=CA	
GA83	GROUP	LNCTL=BSC,	BSC LINE CONTROL
:			
		DIAL=NO	NONSWITCHED LINK
*			
LA83	LINE	ADDRESS=A83, CUTYPE=3271, RETRIES=7,	COMMUNICATIONS ADAPTER LINE ADDRESS BSC 3274 CLUSTER CONTROLLER TRANSMISSION ERROR RECOVERY
:			
		ISTATUS=ACTIVE	VTAM INITIAL STATUS
*			
PA8301	CLUSTER	GPOLL=40, FEATUR2=MODEL2,	GENERAL POLLING ADDRESS 3270 MODEL2 - 1920 CHARACTERS
:			
		ISTATUS=ACTIVE	VTAM INITIAL STATUS
*			
LUA8301	TERMINAL	ADDR=40, MDLTAB=MDLTAB1, MDLENT=BSCENTRY,	TERMINAL POLLING ADDRESS MODEL TERMINAL TABLE NAME MODEL TERMINAL NAME ENTRY
:			
		TERM=3277, ISTATUS=ACTIVE	BSC 3278 TERMINAL TYPE VTAM INITIAL STATUS

Attaching Peripheral Nodes to NCP

This section contains the following subsections:

- Defining Type 2.1 Peripheral Nodes
- Non-Native Network Type 2.1 Connections
- NCP Connections.

Defining Type 2.1 Peripheral Nodes

You should consider the following when defining type 2.1 peripheral nodes.

PUDRPOOL definition statement

NCP must generate a pool of network addresses to establish switched connections to type 2.1 peripheral nodes. If you might add PUs using dynamic reconfiguration, you need to know the maximum number of concurrent type 2.1 peripheral node connections so that you can reserve the appropriate number of control blocks for PU addresses.

Note: The dynamic switched definition facility does not use addresses from the PUDRPOOL.

XMITDLY operand

You can specify the XMITDLY operand on the LINE definition statement to cause the NCP to delay the XID process for the specified period of time to avoid XID contention between the NCP and the type 2.1 peripheral node.

Non-Native Network Type 2.1 Connections

You can connect to type 2.1 physical units that are not in your network, but instead are in a non-native network. You can use the XNETALS start option to network qualify any logical units that you connect to in a non-native network.

If you specify XNETALS=YES, VTAM uses the NETID of the PU (rather than VTAM's network identifier) as the network qualifier of the logical unit's name. For example, in Figure 62, XNETALS=YES and the LU in the attached type 2.1 peripheral node is recognized by VTAM as AS400.LU even though the network identifier of the subarea network is SANETX.

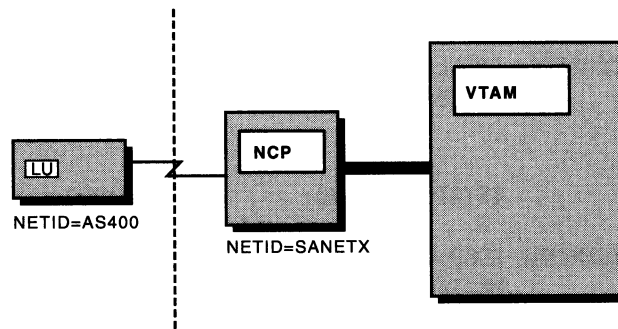


Figure 62. Non-Native Network Type 2.1 Connection

Coding the NETID operand on the PU definition statement is optional (VTAM determines the network ID if not specified). However, if you specify XNETALS=YES, code the NETID operand if you want VTAM to perform dial-out operations to establish a session.

If you specify XNETALS=NO and do not code the NETID operand on the PU definition statement, VTAM assumes all connections are native unless the PU is a dependent LU server (DLUS) PU or a dependent LU requester (DLUR) PU. For information on DLUS and DLUR, see “Dependent LU Server” on page 502.

If you code the NETID operand on the PU definition statement and you code XNETALS=NO (or the XNETALS start option defaults to NO), then the PU's NETID must match VTAM's NETID.

NCP Connections

This section describes the following connections that can be used to attach a peripheral node to NCP:

- Nonswitched SDLC links
- Switched SDLC links
- NCP Token-Ring Interconnection (NTRI) connections
- BSC.

Nonswitched SDLC Connection

You use the GROUP, LINE, PU, and LU definition statements in an NCP subarea node to define a peripheral node attached through a nonswitched SDLC connection. The primary VTAM-only operands that are used for this definition are associated with the following:

- Logon session parameters (DLOGMOD, MODETAB)
- Automatic logon usage (LOGAPPL)

- Unformatted session services facilities (USSTAB, SSCPFM)
- Model terminal support (MDLTAB, MDLENT, ASLTAB, ASLENT)
- Interpret table processing (LOGTAB)
- Pacing control (VPACING)
- Span of control (SPAN).

Following is a sample coding of a nonswitched SDLC connection between an NCP and a peripheral node:

```

G5S3276  GROUP ANS=CONTINUE,      ANS ACTION TOWARD PU
                                DIAL=NO,      NONSWITCHED SDLC LINK
                                LNCTL=SDLC,     LINE CONTROL DISCIPLINE
                                MAXDATA=265,    MAX BYTES PU CAN RCV
                                PACING=1,       NCP-LU PACING OPTIONS
                                SPEED=9600,     9600bps LINE SPEED

:

                                DLOGMOD=T3278M2, (V) DEFAULT LOGMODE ENTRY
                                LOGTAB=LOGONTAB, (V) LOGON INTERPRET TABLE
                                MODETAB=MT3274C, (V) LOG MODE TABLE NAME
                                SSCPFM=USSSCS,   (V) CHAR-CODED MSGS TO SSCP
                                USSTAB=USS3270A, (V) UNFMT SRVCS TABLE NAME
                                VPACING=0        (V) VTAM-NCP PACING VALUE

*
L5032    LINE ADDRESS=(32,HALF), LINE ADDRESS – HALF DUPLEX

:

                                ISTATUS=ACTIVE   (V) VTAM STATUS

*
                                SERVICE ORDER=(C5032A) SERVICE ORDER TABLE

*
P5032A   PU ADDR=C1,             CLUSTER CONTROLLER ADDRESS
                                PUTYPE=2,        PHYSICAL UNIT TYPE

:

                                ISTATUS=ACTIVE   (V) VTAM STATUS

*
LU5032A1 LU LOCADDR=2,

:

                                ISTATUS=ACTIVE

```

Switched SDLC Connection

This section contains the following subsections:

- “Connecting to Non-Native Nodes”
- “Using Error Recovery Activation”
- “Sample Switched SDLC Connection to an NCP”
- “Controlling Dial-In Requests”
- “Controlling Dial-Out Requests”
- “ID Exchange”
- “Controlling Disconnection.”

A switched connection between NCP and a peripheral node can be any of the following:

- An SDLC switched line

- A switched virtual circuit (SVC) of an X.25 packet-switched data network
- A local area network (peripheral nodes only).

Define these connections within switched line groups in the NCP major node, and code either a switched major node or the dynamic switched definition facility to define switched resources to VTAM.

Virtual physical units and logical units with dial-in capability (such as NTO-controlled devices) can be dynamically defined using the configuration services XID exit routine. You can also use PU and LU definition statements in the switched or model major node. Virtual physical and logical units can be coded only for switched lines defined in an NCP major node.

Connecting to Non-Native Nodes: For a VTAM-initiated switched connection to non-native nodes, the XNETALS=YES start option must have been specified; otherwise, session establishment will fail.

Using Error Recovery Activation: Code ANS=CONTINUE on the PU definition statement if you want the LU-LU sessions that traverse this NCP to be continued. (ANS=STOP is the default.) Coding the value of CONTINUE allows an LU-LU session on a switched connection to continue without disruption under either of the following conditions:

- VTAM fails
- The peripheral node is re-activated by VTAM.

Sample Switched SDLC Connection to an NCP: Following is a sample switched major node. You can define multiple sets of switched devices. This enables the VTAM operator to use the VARY command to activate a selected subset of all the switched devices.

```

SWNODE1  VBUILD TYPE=SWNET,
          MAXGRP=2,

:
P60SW1   PU      MAXNO=4
          ADDR=C1,      SDLC ADDRESS OF STATION
          ANS=CONTINUE,  SESSION CONTINUES IF NCP IN SHUTDOWN

:
          IDNUM=08735,   IDENTIFICATION NUMBER
          IDBLK=017,     BLOCK IDENTIFICATION
          ISTATUS=ACTIVE, VTAM INITIAL STATUS
          MAXDATA=256,   MAXIMUM DATA
          MAXOUT=7,      MAXIMUM FRAMES FOR LINK LEVEL RESPONSE
          MAXPATH=2,     MAXIMUM NUMBER OF DIAL OUT PATHS
          MODETAB=MODTAB1, LOGON MODE TABLE
          PACING=7,      BOUNDARY PACING
          PASSLIM=7,
          PUTYPE=2,      SNA CLUSTER CONTROLLER
          SSCPFM=FSS,    USS NOT INVOLVED IN LOGONS
          VPACING=14

*
PATH01   PATH    DIALNO=1234567,  TELEPHONE NUMBER OF PN

:
          GID=5,         SWITCHED LINE GROUP IDENTIFIER

```


	GRPNM=G1SW111, PID=1, REDIAL=3, USE=YES	SWITCHED LINE GROUP NAME DIAL PATH IDENTIFIER NUMBER OF DIAL RETRIES USABLE PATH
*		
L60SLU01 LU	LOCADDR=1,	DEPENDENT LOGICAL UNIT ADDRESS
:		
	ISTATUS=ACTIVE, LOGAPPL=A50PAYRL	INITIAL STATUS ACTIVE CONTROLLING PRIMARY LU

Controlling Dial-In Requests: A station identifier is used by VTAM to identify a physical unit dialing into VTAM on a switched line. The station identifier must be unique for each station in the network.

VTAM uses the station identifier to locate a PU by:

1. Adjacent link station name, if the VERALSID=YES operand is specified on the PU definition statement. The ALS name is constructed from the V0E type F7 received in the XID.
2. The CPNAME and network ID specified on the CPNAME and NETID operands of the PU definition statement.
3. The CPNAME specified on the CPNAME operand of a PU definition statement with no network ID.
4. Station identification number for the PU (IDBLK and IDNUM operands on the PU definition statement). The station identification number is a 48-bit number that is used in the XID exchange during the dial procedure.

When locating a PU, to have VTAM use IDBLK and IDNUM before using CPNAME, code SWNORDER=STATNID on either the LINE or GROUP definition statement or as a start option.

For APPN nodes, the TGN operand can also be specified on the PU definition statement. For details, see Chapter 10, "Implementing an APPN Network" on page 525.

If the peripheral node is a VTAM host defined as a type 2.1 physical unit, use the CPNAME operand; the value of the CPNAME operand must be the same as the value of the SSCPNAME start option in the attaching VTAM type 2.1 peripheral node. The IDBLK and IDNUM operands are ignored.

If the peripheral node is a type 1 or 2 physical unit, use the IDBLK and IDNUM operands on the PU definition statement. VTAM uses the IDBLK and IDNUM values to build the station identifier.

The value that you code for IDBLK is a 3-digit hexadecimal number. Each device type has its own device addressing scheme; you can find the IDBLK value in the component description manuals for the device. Note that this only applies to physical units; do not code IDBLK for logical units.

Because a subarea physical unit can have multiple connections, a connection identifier (IDNUM) is used to distinguish the multiple PU definition statements that might apply to a subarea physical unit.

As soon as a dial-in line has been activated and placed in answer mode (during VTAM startup or by the VTAM operator), the physical unit can dial that line's number and establish communication with VTAM. A physical unit can be defined as call-in only, call-out only, or both.

Controlling Dial-Out Requests: If you want VTAM to dial to a peripheral physical unit on a switched line, you need to give VTAM a path to contact the physical unit. You can identify the path using PATH definition statements. The PATH definition statement must be placed after the PU definition statement to define the paths over which that PU can be contacted. The PATH definition statement identifies to VTAM the line group and the dial characters to be used, as shown in Figure 63 on page 255, parts A and B.

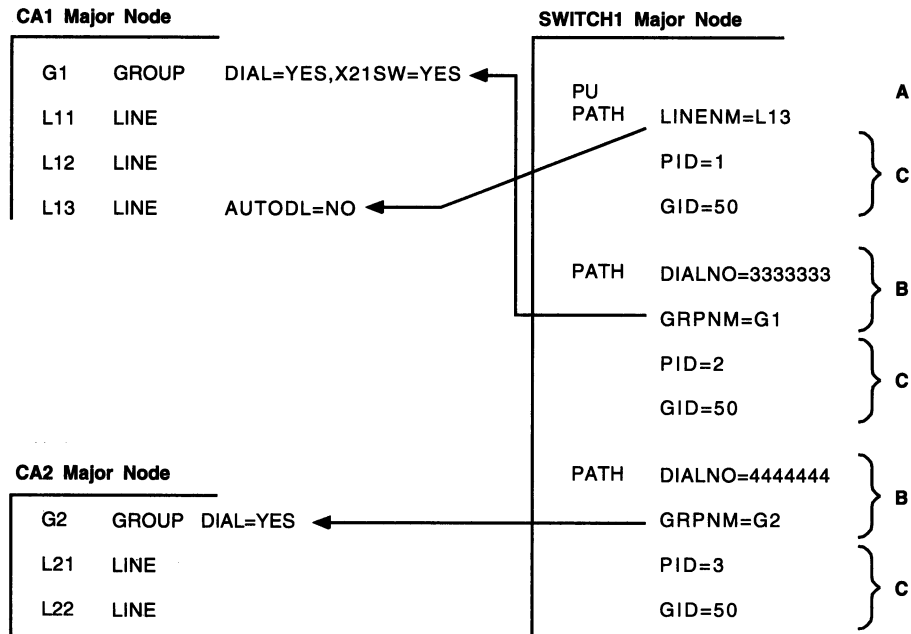
VTAM searches the PATH definition statements for an available path in the order defined. If the line group contains more than one dial-out line, the first available line is tried. If contact cannot be established using that line group, VTAM tries again using the line group identified by the next PATH definition statement. If the line does not have automatic-call capability, the VTAM operator receives a message instructing the operator to manually dial the physical unit.

Two operands on the PATH definition statement, PID and GID, are used to identify each path and to assemble the paths into groups of paths (Figure 63 on page 255, part C). Both operands are used by the VTAM operator. The PID operand identifies the path with respect to its associated physical unit. (The PID is the equivalent of the PATH definition statement's label.) By citing the physical unit name and the PID, the VTAM operator can make that specific path usable or not usable.

The GID operand is used to associate the path with any other path or group of paths within the same switched major node. By naming the GID on the VARY PATH command, the VTAM operator can make that path and all similarly identified paths usable or not usable. The criteria used to assign paths to particular groups are entirely up to you. One possibility is to group paths according to the type of call indicated by the dialing digits (for example, internal (extension) calls, local calls, tie-line or Wide Area Telephone Service (WATS) calls, and long-distance calls).

Dial-out operations for peripheral nodes (terminals) occur when a session is requested with a logical unit whose switched physical unit is not already connected to VTAM.

ID Exchange: Access to LUs can be controlled through their identification. For SNA physical units on switched lines, use SNA ID verification. The IDs of SNA physical units on switched lines are built by VTAM from the PU definition statements. After a physical unit dials into or answers a dial-out request from an application program, an XID (exchange ID) command is sent to the physical unit. The physical unit responds with its station ID, which is sent to VTAM for verification against the IDs defined to VTAM. If the station ID is valid, VTAM sends its SSCP ID to the physical unit in the ACTPU command. If the station ID is not valid, VTAM disconnects the physical unit. When the identifications have been exchanged, normal physical connection procedures continue. This exchange of IDs provides a means for limiting access to the network.



- A These operands supply the essential information needed to create a direct call connection for an X.21 switched line.
- B These operands supply the essential information needed to create a dial-out connection.
- C These operands identify the path (PID) and indicate the path group (GID) with which it is associated.

Figure 63. Major Elements of PATH Definition Statements for Switched Major Node

Controlling Disconnection: You can use the disconnect facility to specify whether VTAM will automatically terminate the SSCP-PU and SSCP-LU sessions when the last LU-LU session is terminated. This is most applicable when the connection is only temporary.

You can also specify a period of time, beginning when the last LU-LU session is terminated, during which VTAM delays termination of the SSCP-PU and SSCP-LU sessions. Delayed disconnection is beneficial if there is usually a minimal time period between the ending of LU-LU sessions and the start of new LU-LU sessions under the same PU.

For a physical unit on a switched link, disconnection means that the dial connection is broken, and the SSCP-PU session is terminated. Therefore, the telephone is hung up, thus potentially saving unnecessary telephone charges. However, the PUs and LUs are not deactivated. They remain in connectable status awaiting another dial-in or dial-out operation, or for a channel-attached resource, awaiting the device to be powered-on. Sessions can be requested with the logical units, and the request simply causes the physical connection to be re-established.

You can enable the disconnect facility by coding the DISCNT operand on the PU definition statement.

If you code DISCNT=YES for a peripheral node, all of the SSCP-LU sessions and the SSCP-PU session for that physical unit are terminated after all LU-LU sessions are terminated. If any of the logical units (or the physical unit) request their own

session termination, VTAM ignores the HOLD part of a character-coded logoff or the LAST-NOTLAST part of a field-formatted Terminate Self request. VTAM also rejects any attempt made by the physical unit to terminate its own SSCP session with a discontact request. If HOLD=YES is specified or defaulted on a LOGOFF request, DISCNT=YES is ignored.

If you specify DISCNT=DELAY for a peripheral node, all of the SSCP-LU sessions and the SSCP-PU session for that physical unit are terminated after all LU-LU sessions are terminated and after the delay period expires. The delay period is determined using the DISCNTIM start option (default is 15 seconds), unless this global limit is overridden by a delay period specified for a particular PU on the DISCNT=DELAY operand or using the MODIFY RESOURCE command.

If you specify DISCNT=NO (the default) for a peripheral node, VTAM terminates the SSCP-LU and SSCP-PU sessions only when one of the following conditions is met:

- VTAM receives a normal discontact request from the physical unit, and all the LU-LU sessions associated with that physical unit have been terminated. All other existing LU-LU sessions are allowed to end normally.
- VTAM receives an immediate discontact request from the physical unit and immediately terminates any existing LU-LU sessions associated with that physical unit. This command overrides any previous normal discontact request from the physical unit.
- All LU-LU sessions have been terminated as a result of one of the following:
 - Character-coded logoff from the SLU with HOLD=NO
 - Terminate self request with LAST specified
 - VARY INACT or TERM command from the operator
 - Cross-domain takedown request
 - TERMSESS macroinstruction.

If any LU-LU sessions were terminated by any other means (for example, by an application program CLSDST macroinstruction), the SSCP-LU and SSCP-PU sessions are not terminated.

The type of processing VTAM performs depends not only on the type of connection but also on whether the type of node attached is peripheral or subarea. For more information on switched connections between subarea nodes, refer to Chapter 7, "Implementing a Subarea Network" on page 389.

The disconnect facility also enables you to specify for the deactivation request of the physical unit whether the "final-use" status is to be sent. VTAM sends the "final-use" status in the Deactivate Physical Unit request if DISCNT=YES,F is specified. If you code NF, "not-final-use" status is indicated. This "final-use" status does not apply when DISCNT=NO, nor does it have any effect on the VARY INACT command.

NCP/Token-Ring Interconnection (NTRI)

Because the token-ring connection is defined to VTAM as though it was a switched SDLC link, code the following:

- NCP major node definitions
- VTAM switched major node.

Also consider the affects of the DISCNT and HOLD operands as discussed in "Controlling Disconnection" on page 255. HOLD=NO should not be specified for token-ring connections since it may cause errors in attached gateways that require operator intervention.

NCP Major Node Definitions: In the NCP major node, define both physical and logical resources:

- The token-ring interface coupler (TIC) connection for NTRI support is defined as a physical connection, which is generated as a nonswitched duplex line.
- The peripheral nodes that are attached to the token ring are defined as logical connections, which are generated as switched half-duplex lines.

The following sample physical and logical definitions are used in an NCP subarea node to attach peripheral nodes to the token-ring. Note that TRNPG1 is the LAN NTRI physical group, and TRNLG1 is the LAN NTRI logical group.

```
TRNPG1  GROUP ECLTYPE=(PHYSICAL,PERIPHERAL) LAN PHYSICAL GROUP
*
TRNL1   LINE ADDRESS=(1088,FULL),          TIC ADDRESS
          LOCADD=400000003023,             LAN ADAPTER ADDRESS
          MAXTSL=1108,                     TRANSMIT MAX DATA (MAX)
          PORTADD=1,                       PHYSICAL PORT ADDRESS
          RCVBUFC=4095,                     RECEIVE MAX DATA

TRNPU1  PU
TRNLU1  LU  ISTATUS=INACTIVE
*
TRNLG1  GROUP ECLTYPE=(LOGICAL,PERIPHERAL), NUMBER DEVICES ON THE LAN
          AUTOGEN=8,                       CONTINUE IF HOST LOST
          ANS=CONTINUE,                     CONNECTION OPTIONS
          CALL=INOUT,                       ASSOCIATE WITH PHYSICAL PORT
          PHYPORT=1

:
COMPOWN=YES
```

TIC Address: The relative address of the TIC on the 3745 communication controller is identified by the ADDRESS operand of the LINE definition statement. The TIC address on the token-ring is defined by the LOCADDR operand of the LINE definition statement and is a locally administered address in this example. Any peripheral node that is attached to the token-ring and needs to communicate through the 3745 TIC must use this address.

Adapter Type: The 3745 communication controller, with NCP V5R3 and subsequent releases, supports a token-ring adapter type 2, which allows the token ring to operate at either 4-megabyte or 16-megabyte speeds. For more detailed information on this support, refer to the *NCP, SSP, and EP Resource Definition Guide*.

Switched Major Node: The peripheral nodes and associated logical units that are attached to the token ring are defined in VTAM using a switched major node.

As shown in the following sample, the same coding that is normally used for an SDLC switched peripheral node is used for a token-ring connection.

```

TRNODE1  VBUILD TYPE=SWNET
P1088TR1 PU      ADDR=04,          SDLC ADDRESS OF STATION
                                CONTINUES IF NCP IN SHUTDOWN (1)
                                ANS=CONTINUE,

:

                                IDNUM=10881,      IDENTIFICATION NUMBER
                                IDBLK=017,        BLOCK IDENTIFICATION
                                ISTATUS=ACTIVE,    VTAM INITIAL STATUS
                                MODETAB=MODTAB1    LOGON MODE TABLE
*
LU108802 LU      LOCADDR=2          DEPENDENT LOGICAL UNIT ADDRESS
*
LU108803 LU      LOCADDR=3          DEPENDENT LOGICAL UNIT ADDRESS

```

For information on the ANS operand and error recovery activation, see “Using Error Recovery Activation” on page 252.

If VTAM is to establish the connection to the token-ring attached peripheral node (for example, use an automatic logon [LOGAPPL] to an application), also code a PATH definition statement in the switched major node.

The DIALNO operand on the PATH definition statement for the callout operation can be used to specify the TIC number, the system access point address, and the token-ring address of the peripheral node.

In the following sample, the TIC number is 02, the system access point address is 04, and the locally administered token-ring address of the peripheral node is the last four digits of the user's telephone extension.

```

PATH01  PATH  DIALNO=0204400000008735,  PERIPHERAL NODE TOKEN-RING
                                                ADDRESS
:
                                GRPNM=TRNLG1      TOKEN-RING LOGICAL GROUP NAME

```

You can also use the DLCADDR operand on the PATH definition statement for the callout operation to specify dial information. Use of DLCADDR allows specification of up to 250 bytes of dial information (versus the 32-character limit of the DIALNO operand). The information from the DIALNO operand above would be coded as follows using the DLCADDR operand. The DLCADDR operands specify the data link control type (TR for token ring), the port address, the DSAP, and the destination address, respectively.

```

PATH01  PATH  DLCADDR=(1,C,TR)
                                DLCADDR=(2,D,2)
                                DLCADDR=(3,D,4)
                                DLCADDR=(4,X,400000008735)

```

Note: DLCADDR information is used by the DLC provider, and VTAM verifies only the syntax of the DLCADDR operand.

Token-Ring Security: You can restrict users that can dial in to your system. The name that you code for the GROUP definition statement in the NCP major node must match the GRPNM operand on the PATH definition statement in the switched major node. Therefore, the value that you code in the GRPNM operand on the PATH definition statement in the switched major node specifies the users in a group that can dial your system. Only the users in the group that you specify on the NCP major node can access your system.

NCP Packet Switching Interface (NPSI) Connections

NPSI is used to connect peripheral nodes to NCP through X.25 switched and virtual circuits and **VSE** X.21 connections.

For NPSI connections, X.21 SHM/MPS is defined on the PATH definition statement using the SHOLD, SHM, and SHMTIM operands. See “Switched Major Node” in the *VTAM Resource Definition Reference* for an explanation of these operands.

X.21 short hold mode/multiple port sharing (SHM/MPS) is supported by coding the SHM and X21SW operands on the GROUP definition statement. Only lines that represent SVCs controlled by NPSI are capable of X.21 SHM/MPS.

You can specify that some lines be used for direct call only. Code AUTODL=NO on the GROUP or PU definition statements in the NCP major node. Then, when VTAM searches a line group for an available line, it cannot select a direct call only line for an auto-call operation.

VTAM does not need to know the telephone numbers. Instead, the PATH definition statement in the switched major node specifies the switched line to be used, and the X.21 network establishes the connection automatically when requested.

If an X.21 direct call line is available, its PATH definition statement should immediately follow the switched physical unit's PU definition statement. This ensures that VTAM attempts to call the physical unit on the direct call line before trying a dial line.

BSC Connection

A binary synchronous (BSC) connection for an NCP-attached peripheral node is defined by the following definition statements in the NCP major node:

- GROUP to define BSC characteristics
- LINE to define the BSC link
- CLUSTER to define the BSC cluster controller
- TERMINAL to define each BSC terminal.

Code PU=YES on the GROUP or LINE definition statement to inform VTAM that the cluster controller and the terminals attached to it are 3270 BSC devices that are to be treated as a physical unit and logical units, respectively.

The FEATUR2 operand describes machine features of the non-SNA device. In the following sample, FEATUR2=MODEL2 indicates that the default screen size for the BSC 3270 is 1920 characters.

```
GA03BSC2 GROUP PU=YES,
              CLOCKNG=DIRECT,
              CRITSIT=YES,
              DIAL=NO,
              DUPLEX=FULL,
              NEGPOLP=.2,
              REPLYTO=23,
              RETRIES=(2,1,2),
:
              TYPE=NCP,
              PAUSE=0,
              POLIMIT=(10,WAIT),
              POLLED=YES,
```

```
LNCTL=BSC
*
LA03288 LINE ADDRESS=(288,40-09)
CA032881 CLUSTER CUTYPE=3271,
          GPOLL=40407F7F
TA03288A TERMINAL ADDR=60604040,
          POLL=40404040,
          TERM=3277
TA03288B TERMINAL ADDR=6060C1C1,
          POLL=4040C1C1,
          TERM=3277
TA03288C TERMINAL ADDR=6060C2C2,
          POLL=4040C2C2,
          TERM=3277
TA03288D TERMINAL ADDR=6060C3C3,
          POLL=4040C3C3,
          TERM=3277
```

You can change the values of the PAUSE and POLIMIT operands using the MODIFY POLL and MODIFY NEG POLL commands respectively. These values can be set as often as you want while the NCP is active. When the NCP is deactivated and later reactivated, the parameter values of the NCP definition are in effect.

MVS,VSE If you define a configuration restart file for the NCP, the parameters for the PAUSE and POLIMIT operands are restored to their checkpointed values, regardless of whether you are using a WARM start for NCP reactivation.

Dynamically Defining Switched Resources

To use the dynamic switched definition facility:

- Write a configuration services XID exit routine, or use the working sample provided with VTAM. This exit routine must be set to build dynamic resources and must be running; otherwise, no dynamic definitions are built, and the device's request for connection is rejected. The exit routine must be written to supply both the name of the resource and the name of the model definition to be used for the PU or LU. The exit routine must also be installed on VTAM for dynamic switched devices support. For information on coding this exit routine, see "Configuration Services XID Exit Routine" in *VTAM Customization*.

VSE A sample configuration services XID exit routine is located in ISTECCS.Z in the VTAM product library.

MVS A sample configuration services XID (ISTECCS ASSEMBLE) exit routine is provided on the sample library (SYS1.SAMPLIB).

VM A sample configuration services XID (ISTECCS ASSEMBLE) exit routine is provided on the SAMPLES (2C2) disk.

- Code one or more model major nodes, or copy and modify the model major node (ISTMODEL) provided on the sample library (SYS1.SAMPLIB). Model major nodes contain the parameters of the PU and LU definitions that VTAM copies to build the dynamic definitions.

You can use the dynamic switched definition facility to dynamically create definitions for any type 1, 2, or 2.1 switched dial-in devices. You cannot dynamically define type 4 or 5 devices. Leased devices that appear to VTAM to be switched

(for example, devices connected by NTRI) can also be dynamically defined using this facility.

You define a model major node with a VBUILD definition statement (TYPE=MODEL). Model PU and LU statements are defined for the minor nodes. These model PU and LU statements are used to create PUs and LUs after the installation exit routine returns the names of the model resource and the previously undefined resource to VTAM.

The PU and LU definition statements in the model major node define the characteristics of switched peripheral nodes. There is no relationship between these PU and LU definition statements in a model major node. The installation exit routine can provide the name of any model PU or LU definition that matches the device's characteristics.

Following is a sample model major node. Notice that you do not need to code LU statements following a PU statement.

MODEL	VBUILD	TYPE=MODEL	MODEL MAJOR NODE
SPUMOD02	PU	ADDR=02, MAXDATA=256, PUTYPE=2, MAXOUT=1, IRETRY=NO, DISCNT=YES, ANS=CONTINUE	PU STATION ADDRESS MAX BYTES PU RECEIVES IN ONE PIU PU TYPE MAX PIUS WITH NO REPLY NCP RETRY DISCONNECT AT SESSION END CONTINUE DURING NCP ANS
SPUMOD01	PU	ADDR=01, MAXDATA=256, PUTYPE=2, MAXOUT=1, IRETRY=NO, DISCNT=YES	PU STATION ADDRESS
SLUMOD01	LU	LOCADDR=1, MODETAB=MODETAB2, PACING=(1,1), VPACING=2	LU LOCAL ADDRESS LOGON MODE TABLE NAME PACING LU AND BOUNDARY NODE PACING VTAM AND BOUNDARY NODE
SLUMOD03	LU	LOCADDR=3, PACING=(1,1), VPACING=2, MODETAB=MODETAB2	LU LOCAL ADDRESS
SLUMOD02	LU	LOCADDR=2, PACING=(1,1), VPACING=2	LU LOCAL ADDRESS

*

A physical unit on a switched line is identified by the unit's CPNAME or IDBLK and IDNUM. As soon as a dial-in line has been activated and placed in answer mode (during VTAM startup or by the VTAM operator), the physical unit can dial that line's number and establish communication with VTAM.

When the connection is broken, the devices are deactivated and the PU and LU definitions in ISTDSWMN are deleted.

Coding DISCNT=YES causes the connection to be broken after all sessions have ended. When deciding whether to code DISCNT=YES on your model PU statements, you need to consider the cost of the line and the number of session starts and ends this particular device will have.

Note: Dynamic definition of switched connections is for dial-in use only. VTAM has no knowledge of the path definitions. Therefore, during failure recovery, even if the device becomes active, VTAM cannot recover the session or use the definition for dial-out operations. The configuration restart files are not updated. Recovery of these resources under takeover situations is not possible.

Attaching Other Domains to VTAM

This section contains the following subsections:

- VTAM-to-VTAM Channel Connections
- VTAM-to-VTAM NCP Connections
- VTAM-to-VTAM External Communication Adapter (XCA) Connections.
- VTAM-to-VTAM Integrated Communication Adapter (ICA) Connections

VTAM-to-VTAM Channel Connections

The following channel connections can be used to connect domains:

- Channel-to-channel connections between host processors using hardware such as a 3088 or 3737 unit
- Multiple channel adapters on a communication controller
- Multipath Channel (MPC) Connections.

Channel-to-Channel Adapter Connection

A channel-to-channel adapter connection between two host processors is defined using a channel-attachment major node. You define two channel-attachment major nodes for each connection. Each channel-attachment major node represents each VTAM's view of the connection. A GROUP definition statement defines the type of links that follow and also carries other operands that sift down to following definition statements that do not explicitly override them.

Code one LINE definition statement for each channel-to-channel adapter. The LINE definition statement defines to VTAM the characteristics of its side of the adapter. Code one PU definition statement for each LINE definition statement.

The definition statements in the following example define the channel link between the two VTAMs shown in Figure 64 on page 264. On the GROUP definition statement, the missing interrupt handler (MIH) operand is coded with a value of YES so that the channel link becomes inoperative after the time period specified on the REPLYTO operand (otherwise, the channel link appears operative, but VTAM cannot use it). The default for the MIH operand is NO.

The LINE definition statement is used to specify the channel unit address (ADDRESS=51A) used in VTAM at one end of the channel link.

The MAXBFRU operand on the LINE definition statement is used to define the number of 4K-byte pages of storage that are used to buffer PIUs for transmission over the channel link. The PIUs are blocked into the 4K pages, transferred over the channel, and then deblocked into VTAM I/O buffers in the other VTAM. The DELAY operand can be used to delay the operation of the data transfer so that more PIUs can be buffered and transferred in a single channel I/O operation. For more information on selecting values for these operands see Chapter 11, "Tuning VTAM for Your Environment" on page 565.

Though the PU definition statement identifies the physical unit type at the other end of the connection to be an NCP (PUTYPE=4), the channel link actually connects to another VTAM, a physical unit type 5. The TGN operand specifies a transmission group number for the channel link transmission group. Parallel transmission groups between two subareas require unique TG numbers so the TGN operand is used to identify the unique transmission group number for this TG.

```

TSCTCAX VBUILD TYPE=CA          CHANNEL-TO-CHANNEL (CTC) CONNECTION
*
CTCGRPX  GROUP LNCTL=CTCA,      CTC CONNECTION
          MIH=YES,              LINK INOPERATIVE - START I/O TIMEOUT
          REPLYTO=10.0,         CHANNEL PROGRAM TIMEOUT

:
          ISTATUS=ACTIVE        VTAM INITIAL STATUS
*
CTCSA51L LINE ADDRESS=51A,      CHANNEL UNIT ADDRESS
          DELAY=.1,             CHANNEL DELAY TIMER
          MAXBFRU=8,            4K PAGE BUFFER FOR CTC DATA TRANSFER

:
          ISTATUS=ACTIVE        VTAM INITIAL STATUS
*
CTCSA51P PU  PUTYPE=4,          PHYSICAL UNIT TYPE = VTAM
          TGN=3,                TRANSMISSION GROUP NUMBER

:
          ISTATUS=ACTIVE        VTAM INITIAL STATUS

```

The capability of having parallel transmission groups allows you to add multiple channel attachments to an adjacent subarea host. With parallel transmission group support, VTAM permits the transmission group number to be specified on the PU definition statement of the channel-attachment major node and on the PATH definition statement. For information about how this affects network routing, see "Parallel Sessions Using Parallel Transmission Groups" on page 312.

Although you can have as many as 255 transmission groups, only 16 of these can be defined because the maximum number of explicit routes that can be defined between two adjacent VTAMs is 16. Each transmission group is single-link capable only. Figure 64 on page 264 illustrates a possible configuration of parallel transmission groups.

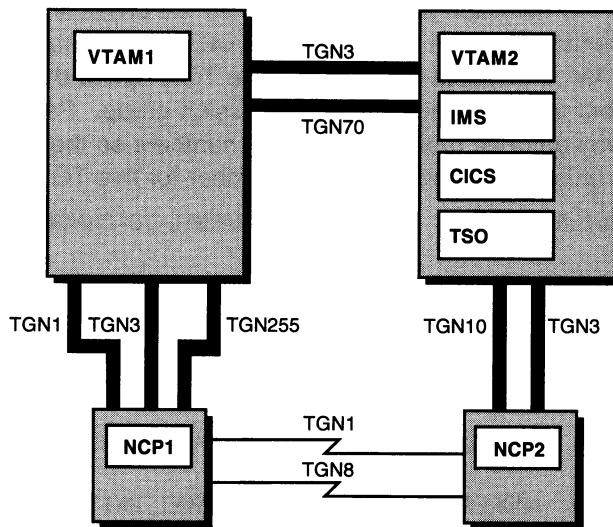


Figure 64. Parallel Transmission Groups in Multiple-Domain Environment with NCP

In the configuration in Figure 64 are two channel-to-channel connections. A LINE and a PU definition statement are required for each of these channel connections between the two VTAMs. The TGN operand for one of the channel connections specifies a transmission group number of 3, as shown in the previous example. The LINE definition statement that defines the other channel link connection must have a TGN operand that specifies a transmission group number of 70. Parallel transmission group support limits each transmission group to a single channel-to-channel link. The two channel link connections (TG3 and TG70) between the VTAMs cannot be defined as a single logical transmission group composed of the two channel links.

Channel-Attached NCP Connection

You can define a channel attachment between a host and an NCP in a channel-attachment major node. This data host attachment eliminates the requirement that the host have ownership (control) of network resources. It provides the ability for VTAM to contact a channel-attached NCP without activating the NCP. By using this definition process, a VTAM data host is not required to establish an SSCP-PU session for data transfer to occur over the channel link to the communication controller. This method of attaching to the network prevents the data host from issuing requests on the SSCP-PU session that can affect the configuration of the local NCP major node or other nodes.

The only definition required in the VTAM data host is a channel-attachment major node definition; no NCP load module, NCP resource resolution table (RRT), or copy of the NCP major node definition is needed. However, the NCP must be loaded and active if VTAM is to contact the NCP and not establish an SSCP-PU session.

The GROUP definition statement defines channel-to-NCP links. It also carries other operands that sift down to following definition statements that do not explicitly override them. When you code LNCTL=NCP, VTAM does not have to have knowledge of the resources attached to the NCP. Code one LINE definition statement for each channel-to-NCP link. The LINE definition statement defines the following to VTAM:

- The name of the link
- The channel unit address of the link

- Certain procedural options to be used for this line.

Code one PU definition statement for each LINE definition statement.

In Figure 64 on page 264, the two channel connections between NCP2 and VTAM2 can be defined using a channel-attachment major node. In this case, VTAM1 is a host that owns all of the network resources attached to both NCP1 and NCP2. VTAM1 loads the NCPs and activates all of the resources.

VTAM2 in this configuration is designated as the data host and thus does not own any network resources. The primary task of the data host (VTAM2) is to support application programs such as IMS, CICS, or TSO. All LU-LU sessions are therefore cross-domain sessions.

The following example illustrates the definition statements used in the VTAM data host (VTAM2) to define one of the channel links between VTAM2 and NCP2. Another LINE and PU definition statement are required to define the other channel link (TG3) between VTAM2 and NCP2.

The LINE definition statement defines the channel unit address for the channel link to the communication controller. The MAXBFRU operand defines the maximum number of I/O buffers that are used by the VTAM host to receive data from the NCP. This operand has the same meaning as the MAXBFRU operand that is specified on the HOST definition statement in the NCP and should contain the same value.

The value of the ISTATUS operand on the LINE definition statement is INACTIVE. If ISTATUS=ACTIVE is specified, the activation process causes VTAM to issue the IST380I message with sense information (081C0010) that is unnecessary. By specifying INACTIVE as the status, the major node activation process completes, and the message is eliminated.

The MAXDATA operand on the PU definition statement also has the same meaning as the MAXDATA operand on the PCCU definition statement. The operand specifies the maximum number of bytes that VTAM can send in a single segment PIU. The TGN operand identifies the unique transmission group number (10) that is associated with this channel link connection.

```

CTSNCP2  VBUILD TYPE=CA                NCP CHANNEL LINK CONNECTION
*
CTSNCP2G GROUP  LNCTL=NCP              NCP CONNECTION
                CHANCON=COND,          CONTACT CONDITIONAL – SUBAREA IN USE

:
                ISTATUS=ACTIVE          VTAM INITIAL STATUS
*
N02CA      LINE  ADDRESS=32A,           CHANNEL UNIT ADDRESS
                MAXBFRTU=32,           MAXIMUM VTAM I/O BUFFERS FOR RECEIVE

:
                ISTATUS=INACTIVE        VTAM INITIAL STATUS
*
N02TG10P PU    PUTYPE=4,               PHYSICAL UNIT TYPE = NCP
                MAXDATA=4096,          MAXDATA TRANSFER
                TGN=10,                TRANSMISSION GROUP NUMBER

:
                ISTATUS=ACTIVE          VTAM INITIAL STATUS

```

In a multiple-domain environment, parallel transmission groups can be defined between VTAM and NCP similar to those in a single-domain environment. This capability allows you to add multiple channel attachments to an NCP subarea. In Figure 64 on page 264, the channel connections between VTAM2 and NCP2 can be defined using channel-attachment major node definitions. In this case, both TG10 and TG3 can be used to communicate with NCP2 without activating NCP1. The transmission group number is specified on the PU definition statement in the channel-attachment major node. Each transmission group is limited to a single link TG. Although you can have as many as 255 transmission groups, only 16 of these can be defined due to the following limitations:

- The number of physical attachments (16 channel adapters) available on a 3745
- The number of explicit routes (16) that can be defined between VTAM and a channel-attached adjacent NCP.

Refer to “Network Routing for Subarea Nodes” on page 151 for information about how this affects network routing.

Note: NCP2 can be activated by the data host if ownership of the NCP2 resources is required as a part of a backup and recovery scenario. This establishes the necessary SSCP-PU session, but it requires that the data host maintain a copy of the NCP major node in its VTAMLST and the NCP resource resolution table (RRT). You might possibly require a copy of the NCP load module at the data host in case the NCP needs to be reloaded and the communication controller does not have a disk (the IBM 3725, for example).

Multipath Channel (MPC) Connections

Multipath Channel (MPC) allows you to code a single transmission group (TG) for host-to-host communication that uses multiple write-direction, read-direction subchannels. The subchannels may be assigned to one or more physical channels based on how you have defined your I/O subsystem to the operating system. Because each subchannel operates in only one direction, the half-duplex turn-around time that occurs with other channel-to-channel connections is reduced.

If you code a TG in which the subchannels are divided between two physical channels, you can increase availability because the TG will have a path to use even if one physical channel is down. Because each TG can use more than one channel, and because the turnaround time required for half-duplex is reduced, throughput is increased.

The subchannels on the physical channel are represented by the subchannel addresses coded on the READ and WRITE operands on the LINE definition statement. One READ subchannel in one host and the corresponding WRITE subchannel in the other host form a complete path. You should consider the ratio of inbound and outbound traffic for a host in deciding how many write subchannels and read subchannels you want to code.

If multiple write subchannels are used, traffic is distributed among the subchannels as traffic requests increase. Arriving data is resequenced before being presented to the application. Priority is assigned to the write subchannels in the order that they are defined.

If you want two TGs, with each TG dividing its subchannels across two parallel physical channels, you have to code two groups, one for each TG. The LINE definition statement for each TG will include subchannel addresses on the READ and WRITE operands from each of the two physical channels.

If a subchannel becomes inoperative, the TG is still active as long as one subchannel is available in each direction. If the last subchannel in either direction becomes inoperative, issue a VARY INACT command, correct the problem (all subchannels must be allocable when activated or activation will fail), and then issue a VARY ACT command to activate the major node.

Note: APPN host-to-host channel connections enable two VTAMs to communicate using APPN protocols over MPC connections. For a description of APPN host-to-host channel connection support, including sample definitions, see “Using Type 2.1 Channel Connections between APPN Nodes” on page 535.

VTAM-to-VTAM NCP Connections

The following NCP connections can be used to connect domains:

- Nonswitched or switched SDLC links between communication controllers
- X.25 permanent or switched virtual circuits using the X.25 NCP Packet Switching Interface (NPSI)
- Token-ring connections using the NCP/Token-Ring interconnection (NTRI).

Nonswitched SDLC Subarea Connection

A nonswitched SDLC link between two NCPs can also be used to connect two VTAM domains. In Figure 65 on page 268, the SDLC connection between NCP1 and NCP2 is used to connect the two VTAM domains (VTAM1 and VTAM2). This connection can operate at whatever speed the communication controller and NCP are capable of supporting. Each end of the SDLC connection is defined in each NCP using NCP definition statements. The connection can be composed of multiple links in the same transmission group because it is a connection between NCPs. To activate the link, one of the following must occur:

- If each VTAM controls its respective NCP (NCP1 and NCP2), each VTAM domain (VTAM1 and VTAM2) must activate its end of the connection.

- If one VTAM controls both NCPs (NCP1 and NCP2), the RNAME operand must be coded on either the PCCU definition statement or the operator command that is used to activate NCP.

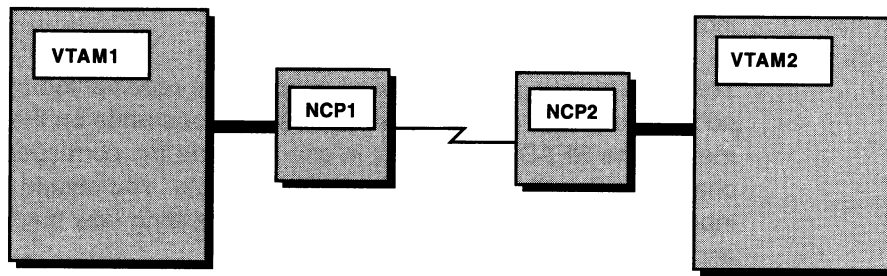


Figure 65. Nonswitched SDLC Subarea Connection

The GROUP and SDLCST definition statements indicate to the NCP the link characteristics that are to be used when the NCP is operating as the primary or secondary end of the link.

The LINE and PU definition statements in the NCP define the link and link station connection between the NCPs. The majority of the operands used to define this connection are associated with NCP. However, you can specify VTAM-only operands such as ISTATUS. For information on the operands that are VTAM-only and are applicable to this connection, refer to “Attaching NCP to VTAM” on page 202.

The sample coding in “Sample Coding for SDLC Primary and Secondary Links and Link Stations” and “Sample Coding for an Intermediate Routing Node Link” on page 269 illustrates the NCP definitions used to define a high-speed connection (1.47 megabytes per second) between NCP1 and NCP2. A set of definitions is required in each NCP.

Sample Coding for SDLC Primary and Secondary Links and Link Stations

PRIM	SDLCST	GROUP=GPRIM, MAXOUT=127, MODE=PRIMARY, PASSLIM=254, RETRIES=(7,3,5),	PRIMARY GROUP NAME LINK LEVEL RESPONSE PRIMARY MODE MAXIMUM PIU SEND COUNT RETRY COUNTS
:			
:		SERVLIM=254	SERVICE ORDER TABLE SCAN
*			
SEC	SDLCST	GROUP=GSEC, MAXOUT=127, MODE=SECONDARY, PASSLIM=254, RETRIES=(7),	SECONDARY GROUP NAME LINK LEVEL RESPONSE SECONDARY MODE MAXIMUM SEND COUNT RETRY COUNT
:			
:		TADDR=C1	SECONDARY POLL ADDRESS
*			
*			
GPRIM	GROUP	LNCTL=SDLC, TYPE=NCP, DIAL=NO, REPLYTO=(1.0,3.0), MODE=PRIMARY,	SDLC LINK CONTROL NCP LINK NONSWITCHED CONNECTION REPLY TIMEOUT PRIMARY LINK STATION


```

:
MODULO=128                                MODULO FOR LINK LEVEL RESPONSE
*
GSEC      GROUP  LNCTL=SDLC,                SDLC LINK CONTROL
              TYPE=NCP,                    NCP LINK
              DIAL=NO,                     NONSWITCHED CONNECTION
              REPLYTO=(NONE,NONE),         NO REPLY TIMEOUT ON SECONDARY
              MODE=SECONDARY,              SECONDARY LINK STATION
              MODULO=128,                  MODULO FOR LINK LEVEL RESPONSE
:
ACTIVTO=120                                ACTIVATE TIMEOUT

```

Sample Coding for an Intermediate Routing Node Link

```

G6INN      GROUP  ACTIVTO=120,              ACTIVATE TIMEOUT
              DIAL=NO,                     NONSWITCHED CONNECTION
              LNCTL=SDLC,                  SDLC LINK CONTROL
              REPLYTO=(10.0,15.0),         RETRY COUNTS
              TYPE=NCP,                    NCP LINK
:
ISTATUS=ACTIVE                            INITIAL VTAM STATUS
*
L51028     LINE   ADDRESS=(1028,FULL),      LINE ADDRESS
              SPEED=1474560,              LINK SPEED
              CLOCKNG=EXT,                 EXTERNAL CLOCKING
              DUPLEX=FULL,                 FULL DUPLEX OPERATION
              PAUSE=(1.0,1.0),             PRIMARY AND SECONDARY PAUSE
              SDLCST=(PRIM,SEC),           SDLCST PRIMARY/SECONDARY
:
ISTATUS=ACTIVE                            INITIAL VTAM STATUS
*
C51028A    PU     ANS=CONTINUE,             CONTINUE AFTER VTAM FAILURE
              PUTYPE=4,                   PHYSICAL UNIT TYPE = NCP
              TGN=8,                      TRANSMISSION GROUP NUMBER
:
ISTATUS=ACTIVE                            INITIAL VTAM STATUS

```

Switched SDLC Subarea Connection

One type of line that can be controlled by an NCP is the switched line. One or more of these lines can be defined as a switched line group in an NCP major node definition. Associated with each line in the NCP major node definition is a PU definition statement to represent a port for the actual physical unit that can be connected to the NCP on that line. The actual physical units must be defined to VTAM in the switched major node.

Note: An automatic calling unit (ACU) is required for switched SDLC subarea connection.

Defining a Switched SDLC Subarea Connection: The PU definition statements in the switched major node define subareas attached over switched lines that can be dialed into or dialed out from a communication controller using NCP. These lines are defined in switched line groups in an NCP major node and a VTAM

switched major node. For dial-out operations, the PATH definition statement defines the possible paths to be used to establish connection between the communication controller or host processor and the other subarea.

Sample Coding for a Switched Major Node for a Switched Subarea: The following coding is an example of definition statements for a switched major node for switched subarea connections:

```

SWN112  VBUILD TYPE=SWNET,          SDLC SWITCHED MAJOR NODE
              MAXGRP=1,              MAXIMUM PATH GROUPS

:
              MAXNO=2                UNIQUE TELEPHONE NUMBERS
*
PU2      PU  ADDR=C1,                PUTYPE4 STATION ADDRESS
              ANS=CONTINUE,          LINK CONNECTION AFTER VTAM FAILURE
              MAXDATA=4096,          MAXIMUM DATA TRANSFER
              SUBAREA=112,           SUBAREA ADDRESS NUMBER
              IDNUM=11201,           IDENTIFICATION NUMBER
              PUYPE=4,               PHYSICAL UNIT TYPE = NCP
              NETID=USIBMCTS,        NETWORK IDENTIFIER

:
              TGN=70                 TRANSMISSION GROUP NUMBER
*
          PATH DIALNO=1234567,        DIAL OUT TELEPHONE NUMBER
              GID=1,                 GROUP IDENTIFIER
              PID=1,                 PATH IDENTIFIER
              REDIAL=2,              RETRY DIAL OPERATION TWICE

:
              GRPNM=GN112SW          NCP GROUP NAME

```

The IDNUM operands associated with both ends of a specific switched link must match for the connection to be established. If there are multiple switched links between two subarea nodes, the IDNUM associated with each link between the subarea nodes must be different for each link. Additionally, the combinations of IDNUM, NETID, and SUBAREA must be unique.

NCP Definition Statements for Switched Subarea Link Connection: The following code illustrates some of the definition statements that are used in the NCP to establish a switched subarea link connection:

```

GN112SW  GROUP  ACTIVTO=480,      ACTIVITY TIMEOUT
                                DIAL=YES,      SWITCHED CONNECTION
                                LNCTL=SDLC,     SDLC LINK CONTROL
                                PUTYPE=4,      PU TYPE 4 (NCP)

:

                                BRKCON=NOWNER   BREAK CONNECTION = NO OWNER (ANS)
*
N112L001 LINE  ADDR=008,          LINK ADDRESS
                                SPEED=9600,     LINK SPEED
                                CLOCKNG=EXT,    EXTERNAL CLOCKING
                                SDLCST=(PRIM,SEC), SDLCST PRIMARY/SECONDARY

:

                                ISTATUS=ACTIVE  INITIAL VTAM STATUS
*
N112P001 PU    PUTYPE=4,          PU TYPE 4 STATION ADDRESS

:

                                ISTATUS=ACTIVE  INITIAL VTAM STATUS

```

These switched connections are established using automatic dial and answer. Either subarea can initiate or terminate the switched connection using a VTAM operator command. Use the VARY DIAL operator command to establish a switched connection. The operator at either VTAM subarea can terminate the switched connection using the VARY HANGUP or VARY INACT operator commands.

Note: A VTAM or NCP capable of switched subarea support cannot communicate across a switched subarea link with an adjacent down-level VTAM or NCP that does not contain switched support.

Controlling Connections: You should also specify ANS=CONT for the switched link so that the connection is not broken and the session can continue. That is, if the NCP enters automatic network shutdown (ANS), the connection is maintained, and session traffic can continue to flow even though the controlling SSCP has lost contact with the NCP. This operand can also be used with the ACTIVTO and BRKCON operands on NCP definition statements to control breaking the switched connection. The ACTIVTO timer value specifies the idle time on the connection before the switched subarea link is disconnected. The BRKCON operand indicates when to initiate the activity timer for the switched connection. You can either enable the inactivity timer when the connection is established (CONNECTO) or start the timer after the NCP completes automatic network shutdown (NOWNER).

Security: Call security verification provides added security for subarea dial operations by providing you with a two-way password verification system without transmitting actual passwords on the network. Each side of the telephone line (the caller and the receiver subarea) generates random data and sends it to the other side. The other side encrypts the random data, using the password as the encryption key, and sends the encrypted data back to the originator, which verifies the received data against its own encryption. The password is defined using the PRTCT operand on the PU definition statement in the switched major node that describes the caller or receiver subarea. The passwords are not transmitted outside of VTAM even when the caller or the receiver subarea is an NCP.

The level of protection provided by call security verification can be equated with a two-way password verification scheme that requires the transmission of each other's passwords. With the call security verification scheme, the possibility of exposing the passwords to a third party on the line is almost entirely eliminated. However, VTAM does not provide any facility to prevent the passwords from printing in the switched major node definition listings. No software password verification scheme can provide full protection from security exposures. For more security, you should use either session-level password verification or session-level cryptography. The PRTCT operand on the PU definition statement in switched major nodes is used to identify the password to be used as the encryption key.

Supporting a Remote Controller: VTAM also supports the use of a switched connection to a link-attached NCP. A switched link can be used to load, activate, or dump the NCP using normal operator commands. No VTAM definition changes are required, but because NCP does not support a password and encryption process, no call security verification is supported. You cannot code the PRTCT password if this support is used to load and activate an NCP over a switched link because a VTAM is required at the other link station to perform the encryption process.

For example, in Figure 66, if VTAM2 fails, VTAM1 can reload and activate NCP2 using a switched connection. However, you cannot use call security verification because VTAM2 is not available to perform the encryption process. Therefore, you cannot code the PRTCT password operand on the switched definition for the PU type 4 connection; if you do, the contact process over the switched link fails.

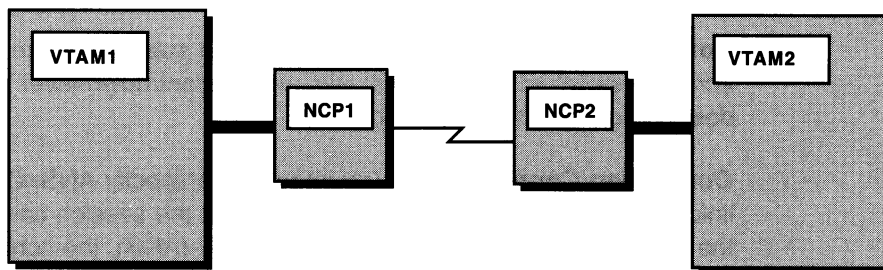


Figure 66. Enhanced Remote Controller Support

Prior to communication, however, information must be exchanged to determine whether the connection can be established. VTAM uses the IDNUM value coded on the PU definition statement to determine if the connection is to a defined NCP subarea node. The IDNUM specification must match the IDNUM specification coded in the NCP on the SWMMT definition statement. For more information on this type of connection, refer to "Enhanced Remote Controller Support" on page 214.

X.25 Subarea Connection (NPSI Support)

Two NCP subarea nodes can communicate across an X.25 packet switching data network (PSDN). Support for an X.25 subarea connection requires the X.25 NCP Packet Switching Interface (NPSI) program product. Both permanent virtual circuit (PVC) and switched virtual circuit (SVC) connectivity options are supported.

When the X.25 NCP Packet Switching Interface (NPSI) receives a virtual call request for a switched connection, VTAM enables X.25 NPSI to identify the caller using call security verification (DTE ID verification). The DTE ID carried on the

incoming call request packet is checked against the DTE IDs defined on the VERID operand of the PATH definition statement in the switched major node. This verification applies to X.21 peripheral and X.25 subarea node connections. For details on X.25 NPSI system generation, refer to *X.25 Network Control Program Packet Switching Interface Planning and Installation*.

Token-Ring Connection (NTRI Support)

A VTAM-to-VTAM subarea node connection can also use a local area network (LAN) token-ring connection between two NCPs. The NCP/Token-Ring Interconnection (NTRI) is used as the LAN connection and is defined to VTAM as a non-switched SDLC link.

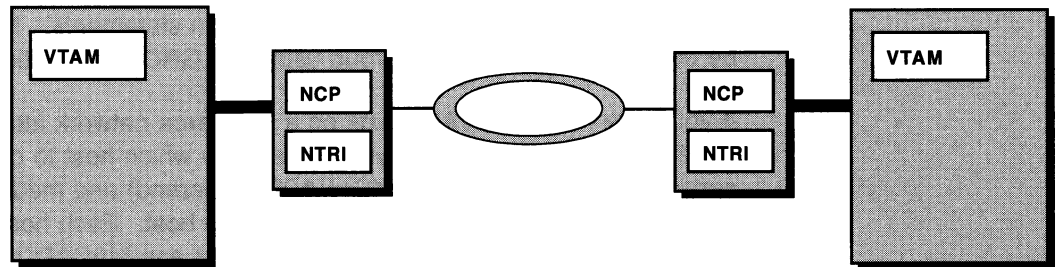


Figure 67. Attaching Domains Using NCP-to-NCP Token-Ring Connection

The following sample shows some of the NCP definition statements that are used in one NCP to define a token-ring connection between two NCP subarea nodes. Define both the physical connection and the logical connection as nonswitched links. Therefore, in contrast to a peripheral node connection, you do not define a VTAM switched major node for a subarea node token-ring connection.

```

N112TGP1 GROUP ECLTYPE=(PHYSICAL,SUBAREA)  TOKEN-RING PHYSICAL GROUP
*
N112TL01 LINE  ADDRESS=(1088,FULL),          TIC ADDRESS
                  LOCADD=400000003023,        LAN ADAPTER ADDRESS
                  MAXTSL=1108,                TRANSMIT MAX DATA (MAX)
                  PORTADD=1,                  PHYSICAL PORT ADDRESS

:
                  RCVBUFC=4095                RECEIVE MAX DATA
*
N112TP01 PU
*
N112TLU1 LU
*
N112TGL1 GROUP ECLTYPE=(LOGICAL,SUBAREA),  TOKEN-RING LOGICAL GROUP
                  PHYPORT=1,                ASSOCIATE WITH PHYSICAL PORT
                  SDLCST=(PRIM,SEC)         SDLCST LINK CHARACTERISTICS
*
N112TR01 LINE
*
N112TP01 PU  ADDR=04400000000030          TOKEN-RING ADDRESS
  
```

You can also specify that a token-ring logical group is for a frame relay connection by coding ECLTYPE=(LOGICAL,FRELAY). This provides a private frame relay network over the token-ring connections without configuring additional physical lines.

VTAM-to-VTAM XCA Connections

A VTAM attached through a 3172 interconnect controller can communicate to other SNA domains through a token-ring, CSMA/CD 802.3, or FDDI local area network (LAN). You define an XCA LAN connection to VTAM using the external communication adapter (XCA) major node.

For each LAN connected through a 3172 interconnect controller, code a VBUILD TYPE=XCA definition statement. The PORT statement identifies the connection between VTAM and the LAN, so you can code only one PORT definition statement for each major node.

A VTAM attached to a LAN through a 3172 interconnect controller is defined to its peer processor by the LINE and PU definition statements. These statements must be coded as a nonswitched group using the GROUP definition statement.

If you have multiple VTAM hosts on a local area network attached through one 3172 interconnect controller, you can specify which host to contact by using the service access point address (SAPADDR operand) and medium access control address (MACADDR operand) to specify the host. Each host should have a different combination of the SAPADDR operand and MACADDR operand.

A VTAM host connected through a 3172 interconnect controller cannot own an NCP across a local area network.

Sample Configuration with a Token-Ring Local Area Network

Figure 68 on page 275 shows a multiple-domain XCA configuration.

To be in session across the XCA connection, application programs in each VTAM must be compatible application programs (that is, they must both be SNA applications programs).

To define the configuration in Figure 68 on page 275, define the following in the VTAM connected through the 3172 interconnect controller:

- One external communication adapter major node to represent the physical unit in the 3172 interconnect controller. This physical unit is used for network management purposes. If you are running the NetView program, it is recommended that you code this definition statement. By defining this physical unit, you can have the 3172 interconnect controller forward the same type of information that a 3174 cluster controller forwards. For example, any alerts detected by the 3172 interconnect controller would be forwarded.
- One external communications adapter major node for each LAN connected to the 3172 interconnect controller.
- Major nodes for any devices connected to the LAN.

For the configuration in Figure 68 on page 275, following is an example of the coding in the VTAM attached through the 3172 interconnect controller:

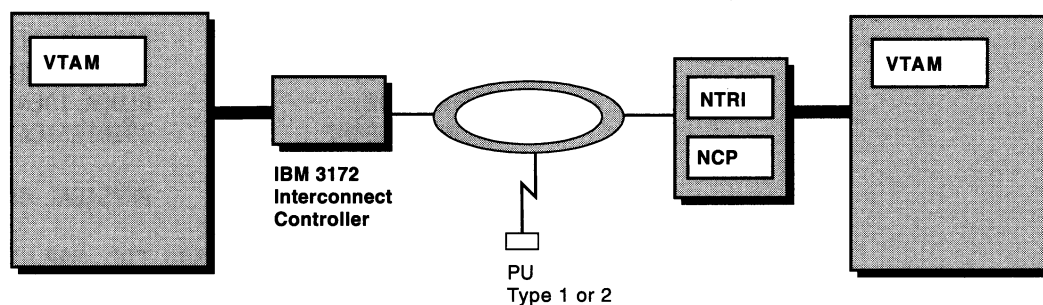


Figure 68. XCA Multiple-Domain Configuration

1. An external communication adapter major node to represent the physical unit in the 3172 interconnect controller:

XCACON1	VBUILD	TYPE=XCA	XCA MAJOR NODE
PORT1	PORT	MEDIUM=BOXMGR,	3172 MANAGER
		CUADDR=500	CHANNEL UNIT ADDRESS
*			
GROUP1	GROUP	ISTATUS=ACTIVE	ACTIVATED AT GEN
LINE1	LINE		
PU1	PU		

2. Another external communication adapter major node for the token ring:

XCACON2	VBUILD	TYPE=XCA	XCA MAJOR NODE
PORT2	PORT	MEDIUM=RING,	TOKEN-RING
		SAPADDR=4,	SERVICE ACCESS POINT ADDRESS
		ADAPNO=1,	ADAPTER NUMBER
		CUADDR=BC0	CHANNEL UNIT ADDRESS
*			
GROUP2A	GROUP	DIAL=YES,	SWITCHED PERIPHERAL NODE
		ANSWER=ON,	DIAL WILL ACCEPT CALLS
		CALL=INOUT,	DIAL IN AND OUT
		ISTATUS=ACTIVE	ACTIVATED AT GEN
LINE2A	LINE	ANSWER=ON	PU CAN DIAL IN
PU2A	PU		
*			
GROUP2B	GROUP	DIAL=NO	NCP SUBAREA
		ISTATUS=ACTIVE	ACTIVATED AT GEN
LINE2B	LINE		
PU2B	PU	MACADDR=400007777757,	LAN MEDIUM ACCESS CONTROL ADDRESS
		PUTYPE=4,	NCP PU
		SAPADDR=8,	SERVICE ACCESS POINT ADDRESS
		SUBAREA=22,	NCP'S SUBAREA NUMBER
		TGN=1	TRANSMISSION GROUP NUMBER

3. A switched major node for the peripheral node attached to the LAN:

SWNODE1	VBUILD	TYPE=SWNET	
PU2	PU	DISCNT=YES, DYNLU=YES, IDBLK=002, IDNUM=000002 MAXPATH=5, PUTYPE=2,	DISCONNECT FACILITY DYNAMIC DEFINITION OF ILU (CDRSC) BLOCK IDENTIFICATION IDENTIFICATION NUMBER PHYSICAL UNIT TYPE
*			
LANPATH2	PATH	DIALNO=0104000069010C59, GRPNM=GROUP2A	TIC, SAP, MACADDR LOGICAL GROUP DEFINITION
LU2	LU	LOCADDR=2, ISTATUS=INACTIVE	DEPENDENT LU INITIAL STATUS INACTIVE

Notes:

1. You need to code one LINE and PU definition statement in the dial GROUP definition statement for every simultaneous peripheral node connection.
2. Switched devices connected to a LAN are defined to VTAM using the switched or model major node. See "Switched SDLC Connection" on page 235, "Defining Switched Resources Dynamically" on page 143 and "Dynamically Defining Switched Resources" on page 260.
3. Multiple VTAM hosts can be in session with different stations on the LAN through one 3172 interconnect controller. To implement this support, code a different SAPADDR and MACADDR value combination to distinguish between the two hosts.

Sample Configuration with CSMA/CD 802.3 Local Area Network

Figure 69 shows a multiple-domain XCA configuration with a CSMA/CD 802.3 local area network and two 3172 interconnect controllers.

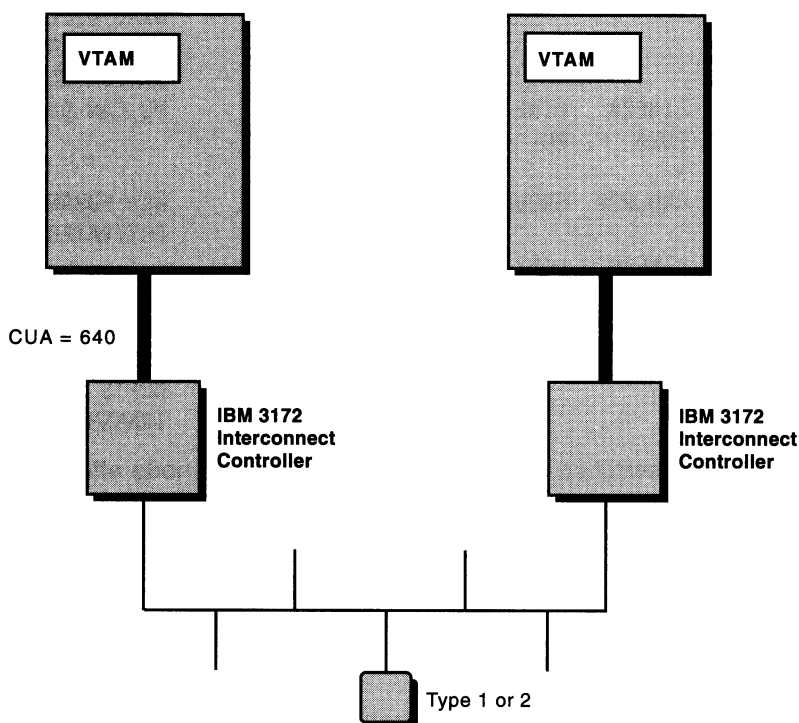


Figure 69. XCA Multiple-Domain Configuration with CSMA/CD

To define the configuration in Figure 69, define the following in the first VTAM:

- One external communication adapter major node to represent the physical unit in the 3172 interconnect controller
- One external communication adapter major node for each LAN connected to the 3172 interconnect controller
- Major nodes for any devices connected to the LAN.

Following is the sample code for the configuration in Figure 69 on page 276:

1. Define an external communication adapter major node to represent the physical unit in the 3172 interconnect controller:

```
XCACON1  VBUILD TYPE=XCA          XCA MAJOR NODE
PORT1    PORT  MEDIUM=BOXMGR,      3172 MANAGER
          CUADDR=C04                CHANNEL UNIT ADDRESS

*
GROUP1    GROUP ISTATUS=ACTIVE      ACTIVATED AT GEN
LINE1     LINE
PU1       PU
```

2. Define another external communication adapter major node for the CSMA/CD 802.3 local area network:

```
XCACON2  VBUILD TYPE=XCA          XCA MAJOR NODE
PORT2    PORT  MEDIUM=CSMACD,      CSMA/CD 802.3
          SAPADDR=4,                SERVICE ACCESS POINT ADDRESS
          ADAPNO=1,                  ADAPTER NUMBER
          CUADDR=640                 CHANNEL UNIT ADDRESS

*
GROUP2A   GROUP DIAL=YES,           SWITCHED PERIPHERAL NODE
          ANSWER=ON,                 DIAL WILL ACCEPT CALLS
          CALL=INOUT,                DIAL IN AND OUT
          ISTATUS=ACTIVE              ACTIVATED AT GEN
LINE2A    LINE ANSWER=ON             PU CAN DIAL IN
PU2A      PU

*
GROUP2B   GROUP DIAL=NO             SUBAREA
          ISTATUS=ACTIVE              ACTIVATED AT GEN
LINE2B    LINE
PU2B      PU  MACADDR=02608C1C0A21  LAN MEDIUM ACCESS CONTROL ADDRESS
          PUTYPE=5,                   VTAM HOST
          SAPADDR=4,                   SERVICE ACCESS POINT ADDRESS
          SUBAREA=2,                   HOST'S SUBAREA NUMBER
          TGN=2                        TRANSMISSION GROUP NUMBER
```

3. Define a switched major node for the peripheral node attached to the LAN:

```
SWNODE1  VBUILD TYPE=SWNET
PU2       PU  DISCNT=YES,            DISCONNECT FACILITY
          IDBLK=002,                  BLOCK IDENTIFICATION
          IDNUM=000002                 IDENTIFICATION NUMBER
          MAXPATH=5,
          PUTYPE=2,                     PHYSICAL UNIT TYPE

*
LANPATH2 PATH DIALNO=010402608C1C0B35, TIC, SAP, MACADDR
GRPNM=GROUP2A  LOGICAL GROUP DEFINITION
LU2           LU  LOCADDR=2,           DEPENDENT LU
          ISTATUS=INACTIVE             INITIAL STATUS INACTIVE
```

VTAM-to-VTAM ICA Connections (VM, VSE)

The following ICA connections can be used to connect domains:

- Nonswitched or switched SDLC link attached to the communication adapter
- Token-ring attached to the ES/9000 series processor
- X.25 permanent or switched virtual circuit
- **VSE** X.21 switched or nonswitched connections.

You can use any of the above connections with the ES/9000 series processor.

Each connection is limited to a single-link transmission group.

Parallel transmission groups provide the capability to define more than one link between the adjacent subareas. In Figure 70, the two links connecting the VTAMs are defined as two single-link transmission groups (TG3 and TG70). The parallel transmission group connections can also be cross-domain links that are connected to an NCP that is controlled by another VTAM domain. Parallel transmission groups between adjacent subareas require unique transmission group numbers. For example, Figure 70 uses TGN3 and TGN70. The TG numbers need to be unique between two adjacent subarea nodes regardless of the type of connection (SDLC, token-ring, X.25).

Note: In an environment with an ES/9000 series processor, the channel-attachment major node can be used to define channel-to-channel and channel-attached NCP connections. For more information about these connections refer to “Attaching Other Domains to VTAM” on page 145.

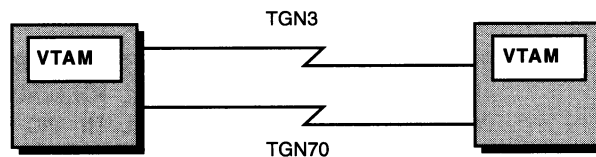


Figure 70. Parallel Transmission Groups in Multiple-Domain Environment with ES/9000 Series Processor

Nonswitched SDLC Subarea Connection

The GROUP definition statement defines an SDLC nonswitched line group. It also carries other operands that sift down unless you code PU and LU definition statements to override them explicitly. If a line is defined in more than one line group, only one of the groups can be active at a time.

Code one LINE definition statement for each SDLC nonswitched line.

Code the PU definition statement for each subarea node with which VTAM communicates over this nonswitched SDLC link.

VTAM uses the PU definition statement to define the link station for an adjacent host processor (physical unit type 5) or communication controller (physical unit type 4).

In the following sample, the SDLC nonswitched link is used to connect an ES/9000 series processor to the NCP that is controlled by a VTAM in another domain:

```

A80ICA  VBUILD TYPE=CA          CHANNEL-ATTACHMENT
A80SNAGR GROUP LNCTL=SDLC,      SDLC LINK CONTROL
                                   NONSWITCHED CONNECTION

:
                                   ISTATUS=ACTIVE      INITIAL VTAM STATUS
*
A80SNALN LINE ADDRESS=A80,     CHANNEL ADDRESS
                                   RETRIES=7,          ERROR RETRY COUNT

:
                                   ISTATUS=ACTIVE      INITIAL VTAM STATUS
*
A80SNAPU PU  PUTYPE=4,         ADJACENT SUBAREA = NCP
                                   SUBAREA=109,        ADJACENT SUBAREA ADDRESS
                                   TGN=3,             TRANSMISSION GROUP NUMBER

:
                                   ISTATUS=ACTIVE      INITIAL VTAM STATUS

```

Switched SDLC Subarea Connection

In a multiple-domain environment, ES/9000 series domains can be connected using switched SDLC links.

These lines are defined in switched line groups in a channel-attachment major node for the communication adapter hardware facilities and a VTAM switched major node for the logical connection. The PU definition statement defines the adjacent subarea that is attached using the switched line. The switched connection can be established using a dial-in or dial-out for an ES/9000 series communication adapter.

For dial-out operations, the PATH definition statement defines the possible paths to be used to establish connection between an ES/9000 series processor and the other host processor or communication controller.

The following sample shows the definition statements for a switched major node for switched subarea connections in an environment with an ES/9000 series processor:

```

SWN112  VBUILD TYPE=SWNET,      SDLC SWITCHED MAJOR NODE
                                   MAXGRP=1,          MAXIMUM PATH GROUPS

:
                                   MAXNO=2            UNIQUE TELEPHONE NUMBERS
*
PU2      PU  ADDR=C1,           PU TYPE 4 STATION ADDRESS
                                   ANS=CONTINUE,       LINK CONNECTION AFTER VTAM FAILURE
                                   MAXDATA=265,        MAXIMUM DATA TRANSFER
                                   SUBAREA=111,        SUBAREA ADDRESS NUMBER
                                   IDNUM=11101,       IDENTIFICATION NUMBER
                                   PUYE=4,           PHYSICAL UNIT TYPE = NCP

:
                                   TGN=3              TRANSMISSION GROUP NUMBER
*
          PATH DIALNO=1234567,  DIAL OUT TELEPHONE NUMBER
                                   GID=1,             GROUP IDENTIFIER
                                   PID=1,             PATH IDENTIFIER
                                   GRPNAME=GSW101,    GROUP NAME

```

```

:
                                REDIAL=2                RETRY DIAL OPERATION TWICE

```

The following sample shows the corresponding channel-attachment major node definition that defines the physical connection on the communication adapter. The SUBADIAL=YES operand must be coded for a switched subarea node connection in addition to the DIAL=YES operand, which defines the connection as a switched SDLC link.

```

SW101  VBUILD TYPE=CA                SDLC SWITCHED CONNECTION
*
GSW101  GROUP  LNCTL=SDLC,            SDLC LINE CONTROL
                DIAL=YES,              SWITCHED CONNECTION
                SUBADIAL=YES,          SUBAREA DIAL CONNECTION

:
                                CALL=INOUT            DIAL-IN/OUT OPERATION
*
LSW101A LINE  ADDR=030,              PU TYPE 4 CHANNEL ADDRESS
                ACTIVTO=60,           FRAME NOT RECEIVED TIMEOUT VALUE
                REPLYTO=1.0,          REPLY TIMEOUT
                RETRIES=7,            RETRY ERROR COUNT

:
                                ISTATUS=ACTIVE        VTAM INITIAL STATUS
*
PSW101A PU    ISTATUS=ACTIVE        VTAM INITIAL STATUS

```

X.25 Subarea Connection

In an environment with an ES/9000 series processor, a VTAM subarea node can attach to another VTAM subarea node or to a remote NCP subarea node across an X.25 packet switching data network (PSDN). For each node type (type 4 NCP or type 5 VTAM), both switched virtual circuit (SVC) and permanent virtual circuit (PVC) connectivity is supported. Figure 71 shows a sample X.25 configuration.

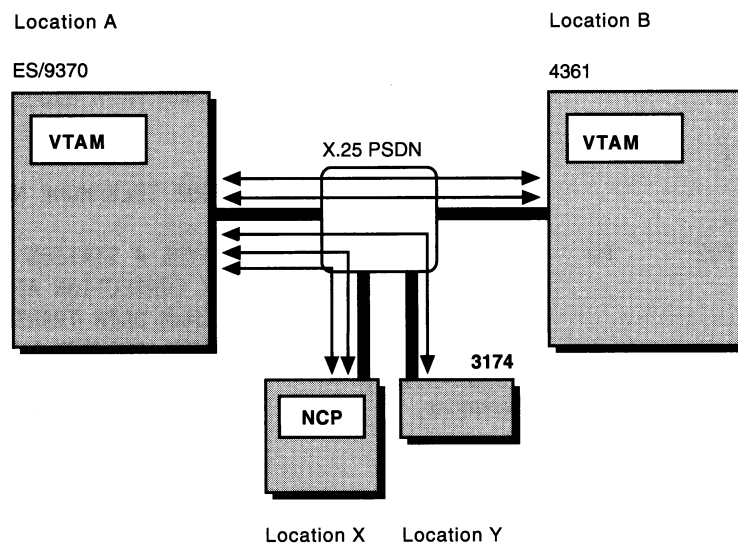


Figure 71. X.25 Configuration in Multiple-Domain Environment

Support for the type 4 connection requires the X.25 NCP Packet Switching Interface (NPSI) program product. Refer to *X.25 Network Control Program Packet Switching Interface Planning and Installation* for details on NPSI system generation.

The following definitions for location A include multiple transmission group support and call security verification (DTE ID verification) definition requirements. For call-out operations, the PATH definition statement identifies the remote network address where a connection is to be established. When VTAM receives a virtual call request for a switched connection, identification of the caller is possible using call security verification (DTE ID verification). The DTE ID carried on the incoming call request packet is checked against the DTE IDs defined on the VERID operand of the PATH definition statement in the switched major node. This verification applies to X.21 peripheral and X.25 subarea node connections. When VERIFY=OUT or VERIFY=INOUT is coded, all incoming calls are verified and rejected if the VERID values do not match.

```

* *****
* CALOCA25 - PACKET MAJOR NODE DEFINITION AT LOCATION A
* *****
CALOCA25 VBUILD TYPE=PACKET X.25 PORT
          PORT CUADDR=(800,808), X.25 CHANNELS
          NETTYPE=3,
          VCALLS=(,004,006,,), SVC RANGE, INOUT CIRCUITS
          PLENGTH=256, SVC PKT SIZES (NONSTD)
          PWINDOW=7, SVC WINDOW SIZE (NONSTD)
          CHARGACC=YES REVERSE CHARGING ACCEPTED
          VCPARMS LC=(1,2),
          PLENGTH=256, PVC PACKET SIZE (NONSTD)
          PWINDOW=7 PVC WINDOW SIZE (NONSTD)
GROUP1 GROUP LNCTL=SDLC, PVC LINE GROUP
          DIAL=NO,
          ISTATUS=INACTIVE
LINE001 LINE ADDRESS=01 ES/9000 NODE AT LOCATION B
PU001 PU PUTYPE=5, TYPE 5 PHYSICAL UNIT
          SUBAREA=5, SUBAREA ADDRESS
          TGN=2 TRANSMISSION GROUP NUMBER
LINE002 LINE ADDRESS=02 NCP/NPSI NODE AT LOCATION X
PU002 PU PUTYPE=4, TYPE 4 PHYSICAL UNIT
          SUBAREA=255, SUBAREA ADDRESS
          TGN=2 TRANSMISSION GROUP NUMBER
GROUP2 GROUP LNCTL=SDLC, SVC LINE GROUP
          DIAL=YES,
          ANSWER=ON
LINE003 LINE USED FOR 3174 AT LOCATION Y
PU003 PU
GROUP3 GROUP LNCTL=SDLC, SVC LINE GROUP
          DIAL=YES,
          ANSWER=ON, CALL IN/OUT
          CALL=INOUT,
          SUBADIAL=YES
LINE004 LINE USED FOR SVC TO ES/9000 OR NCP/NPSI
PU004 PU
LINE005 LINE USED FOR SVC TO ES/9000 OR NCP/NPSI
PU006 PU

```

Attaching Other Domains to VTAM

```
* ***** *
* SWLOCA01 - SWNET MAJOR NODE DEFINITION AT LOCATION A *
* ***** *
SWLOCA01 VBUILD TYPE=SWNET, SWITCHED MAJOR NODE
                      MAXGRP=3,
                      MAXNO=3
SWPU004 PU ADDR=C1,
                      PUTYPE=2,
                      MAXPATH=1,
                      MAXDATA=262,
                      IDBLK=017,
                      IDNUM=70933
                      PATH DIALNO=(47031205), CALLED DTE ID LOC Y
                      GRPNM=GROUP2 REFER TO SVC GROUP NAME
SWLU004 LU LOCADDR=2,
                      DLOGMOD=D4C32762,
                      LOGAPPL=CICS,
                      PACING=1,
                      VPACING=2
SWPU005 PU IDNUM=00001,
                      MAXDATA=256,
                      SUBAREA=255, NCP SUBAREA ID LOC X
                      TGN=1,
                      PUTYPE=4, NCP PU TYPE 4
                      MAXPATH=5
                      PATH DIALNO=(47031206), CALLED DTE ID AT LOC X
                      GRPNM=GROUP3 REFER TO SUBAREA GROUP ID
                      VERIFY=IN, DTE ID CALL VERIFICATION
                      VERID=(47031204) ACCPT DTE ID FROM LOC A
SWPU006 PU IDNUM=00002,
                      MAXDATA=256,
                      SUBAREA=005, VTAM SUBAREA ID LOC B
                      TGN=1,
                      PUTYPE=5, VTAM PU TYPE 5
                      MAXPATH=5
                      PATH DIALNO=(47031207), CALLED DTE ID AT LOC B
                      GRPNM=GROUP3 REFER TO SUBAREA GROUP ID
                      VERIFY=IN, DTE ID CALL VERIFICATION
                      VERID=(47031204) ACCPT DTE ID FROM LOC A
```

DTE-to-DTE Connections

You do not need an X.25 Packet Switched Data Network (PSDN) between two DTEs. With DTE-to-DTE support, one of the DTEs can act as a DCE on any ES/9000 series TCS X.25 port attached through a leased line. You can connect the two DTEs with two modems or an intervening modem eliminator.

To enable this support, you must use the NETTYPE definition statement. NETTYPE specifies the type of network to which the device wants to connect. For example, if you code NETTYPE=DCE, that means the device wants to connect to a device that will act as a DCE.

You must specify NETTYPE=DTE on the PORT statement of the packet major node for VTAM to assume the DCE role.

When you code NETTYPE=DTE, you must code the VCALLS operand of the PORT definition statement from the perspective of the remote DTE. This means

that the incoming channels that you code will be used as outgoing channels by this DTE.

Figure 72 is a multiple-domain DTE-to-DTE configuration.

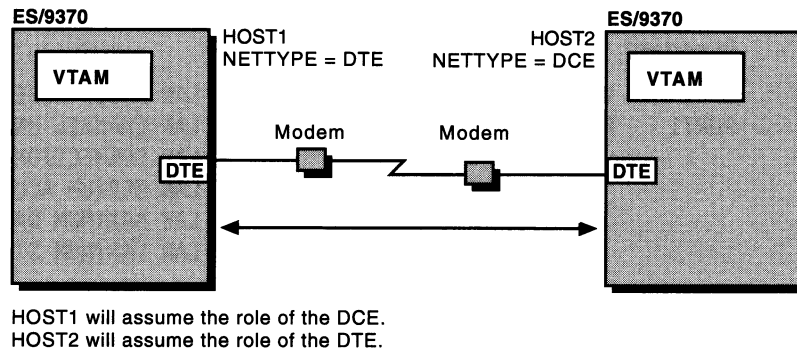


Figure 72. DTE-to-DTE Multiple-Domain Configuration

Note: You cannot string together more than two VTAMs, and there is no support for multiple-network routing.

Token-Ring Connection

An ES/9000 series VTAM domain can communicate with other domains using a token-ring local area network (LAN) connection. The token-ring LAN connection is defined in an ES/9000 series processor using a VTAM LAN major node.

For a token-ring connection, you must code a VBUILD definition statement to identify the LAN to VTAM. Code a PORT definition statement to identify the connection between VTAM and the LAN. Only one PORT definition statement is coded for each VBUILD TYPE=LAN definition statement. Coding the PORT definition statement causes the LNCTL operand of the GROUP definition statement that follows to use the default of LNCTL=SDLC. The PORT definition statement identifies the address of the access protocol group in the Token-Ring Subsystem of the ES/9000 series processor. Its operands define parameters that are passed to the subsystem.

The values of the MACADDR, SAPADDR, and CUADDR operands are used to route information between the LAN and VTAM. When information flows from VTAM down to the LAN, the first digits of the CUADDR operand define the channel used to route the information to the subsystem. When information flows from the LAN up to VTAM, the MACADDR operand is used to route the information to the subsystem. The subsystem builds a table from the SAPADDR and CUADDR operands and uses the table to route the information to the channel unit address (CUA) corresponding to the service access point (SAP).

An ES/9000 series processor attached to the token-ring as a peer processor is defined to VTAM by the LINE and PU definition statements. The LINE and PU definition statements must be specified as pairs in a nonswitched line group with a GROUP definition statement. The GROUP definition statement defines a line group for ES/9000 series processors attached to the ring that is attached to VTAM.

The LINE definition statement defines the following:

- Each peer processor on the token-ring
- The name of the line

Sharing NCP Peripheral Nodes (CMC Configurations)

- Procedural options to be used for this line.

You must code a PU definition statement for each ES/9000 series processor with which VTAM communicates over this token-ring.

Following is an example of how a connection between ES/9000 series processors is coded:

```
LAN1      VBUILD TYPE=LAN          LAN MAJOR NODE
PORT1     PORT  CUADDR=840,         LAN CHANNEL UNIT ADDRESS
          LANCON=(10.1,3),          LAN CONNECTION VALUES
          MACADDR=00000031,         LAN MEDIUM ACCESS CONTROL ADDRESS
          MAXDATA=2012,             LAN MAXIMUM DATA LENGTH
          MAXSTN=8,                 LAN MAXIMUM STATIONS

:
          SAPADDR=4                 LAN SERVICE ACCESS POINT ADDRESS
GP1LAN    GROUP DIAL=NO,            LEASED (PEER 9000 SERIES) ATTACHMENT
          ISTATUS=INACTIVE

LN1LAN    LINE
PU1LAN    PU      ISTATUS=ACTIVE,    LAN ACKNOWLEDGMENT VALUES
          LANACK=(5.0,2),           LAN CONNECTION VALUES
          LANCON=(10.0,5),          LAN INACTIVITY VALUE
          LANINACT=25.5,            LAN RESPONSE VALUES
          LANRESP=(25.4,9),         LAN WINDOWING VALUES
          LANSDDW=(7,1),            LAN MEDIUM ACCESS CONTROL ADDRESS
          MACADDR=400000000032,

:
          SAPADDR=4                 LAN SERVICE ACCESS POINT ADDRESS
```

Sharing NCP Peripheral Nodes (CMC Configurations)

This section contains the following subsections:

- Channel-Attached NCPs in a CMC Configuration
- Link-Attached NCPs in a CMC Configuration
- Controlling Resource Ownership in a CMC Configuration
- Problem Determination in a CMC Configuration.

Channel-Attached NCPs in a CMC Configuration

Several operands on the PCCU definition statement are used to define channel characteristics. If you do not define a channel in a channel-attachment major node, the values on the PCCU definition statement are used because the channel is implicitly defined and activated automatically when the NCP is activated.

However, when you define the channel in a channel-attachment major node, the definition statements in the channel-attachment major node contain operands similar to those on the PCCU definition statements. These operands are the following:

- CHANCON=**CONDIUNCOND**

For a data host, the CHANCON operand on the PCCU definition statement is replaced by the CHANCON operand on the LINE definition statement in the channel-attachment major node.

- **CUADDR**=*channel device address*

For a data host, the CUADDR operand on the PCCU definition statement is replaced by the ADDRESS operand on the LINE definition statement in the channel-attachment major node.

- **MAXDATA**=*size*

For a data host, the MAXDATA operand on the PCCU definition statement is replaced by the MAXDATA operand on the PU definition statement in the channel-attachment major node.

- **DELAY**=*time*

For a data host, the DELAY operand on the PCCU definition statement is replaced by the DELAY operand on the PU definition statement in the channel-attachment major node.

Activating the channel-attachment major node to activate the channel causes VTAM to use the values defined in the channel-attachment major node. Later, if NCP1 is activated, these same operands, if they appear on NCP1's PCCU definition statements, are ignored. If you are defining the channel in a channel-attachment major node and want to avoid confusion concerning these operands, avoid coding duplicate values on the PCCU definition statement.

The RNAME operand on the PCCU definition statement can name a link station associated with a channel link defined in a channel-attachment major node. When the host activates each NCP, the link station and link defined in the channel-attachment major node representing the channel are activated automatically.

The values defined on the PU definition statement in the channel-attachment major node are used for the CUADDR, CHANCON, DELAY, and MAXDATA operands. The PCCU values for these operands are used only when the channel-link station is not defined in a channel-attachment major node.

A channel-attached NCP requires one HOST definition statement for each host that activates the NCP. The HOST definition statement in the NCP major node has a MAXBFRU operand for defining the number of buffers that the host can use for receiving data from the NCP. If you do not define the channel in a channel-attachment major node, the MAXBFRU value on the HOST definition statement is used because the channel is implicitly defined and activated automatically when the NCP is activated.

However, when you define the channel in a channel-attachment major node, the MAXBFRU operand on the LINE definition statement for that major node defines the number of buffers the host can use for receiving data. Figure 36 on page 146 shows an example of this. When HOST1 activates its channel-attachment major node, VTAM uses the value of the MAXBFRU operand on the LINE definition statement. Later, if NCP1 is activated by HOST1, the MAXBFRU value on the HOST definition statement is ignored.

When you define the channel in a channel-attachment major node, VTAM also ignores the MAXBFRU value on the HOST definition statement when the link station for the channel is activated automatically. This is because the value is determined by the RNAME operand when the NCP is activated.

Recommendation: The MAXBFRU operands on the LINE definition statement of the channel-attachment major node should contain the same values as the

MAXBFRU operand specified on the HOST definition statement of the NCP major node.

For more information on defining a channel-attachment connection between an NCP and a VTAM data host, refer to “Channel-Attached NCP Connection” on page 264.

Link-Attached NCPs in a CMC Configuration

In a multiple-domain network, you can define a switched major node in one domain and treat its logical units as cross-domain resources in other domains. You can also define a switched major node in each domain that owns an NCP and has a dial port for a terminal. If defined in more than one domain, NCPs can dial into or be dialed out from each of the host processors in which the major node is defined.

No HOST definition statement should be defined for a link-attached NCP.

Controlling Resource Ownership in a CMC Configuration

Using the OWNER operand on the PCCU definition statement enables you to easily identify and coordinate ownership of NCPs. You can also specify ownership of resources at the physical unit (PU) level for peripheral nodes attached to an SDLC link. You can share ownership of resources on a line and ownership of the lines. In addition, the use of the OWNER operand has implications in backup and recovery plans involving shared NCPs.

The acquisition and release of resources of a shared NCP allow more selectivity in specifying which resources are affected. The VARY ACQ and VARY REL operator commands, in conjunction with the OWNER support, enable you to selectively acquire and release resources. The VTAM operator can specify VARY ACQ,OWNER=sscp name,ID=ncp name to specify that the VARY ACQ command is to apply only to those resources defined as belonging to the specified owner. An equivalent VARY REL operator command allows the VTAM operator to selectively release ownership of a resource.

The TYPE=GIVEBACK operand on the VARY REL operator command permits the VTAM operator to release ownership of the resource without disrupting an LU-LU session. However, because this function operates only at the line level, all physical units attached to that line also have the ownership terminated without disrupting LU-LU sessions. If a host wants to relinquish ownership of the logical unit, it releases or deactivates the logical unit and its associated physical unit.

Note: You can release ownership of an SNA resource without disrupting an LU-LU session.

Every resource attached to the NCP is owned by a VTAM host. The values you code for each NCP major node determine which host owns which resources. The value of the OWNER operand on the PCCU definition statement is matched with the OWNER operands on the GROUP, LINE, and PU definition statements in the NCP.

More than one host can share ownership of a line or a PU type 4 link station. You specify this in the NCP by coding the appropriate names as values of the OWNER operand on the GROUP or LINE definition statements. In this situation, the first host to activate the physical unit becomes its owner. Other hosts cannot activate

this physical unit unless ownership is relinquished. In addition, ownership of PUs can be specified by the OWNER operand on the PU definition statement.

Use the BACKUP operand on the PCCU definition statement to allow a backup host to take over NCP resources when there is a failure in the host that owns them. Code BACKUP=YES to identify all resources that this host can take over in case of a failure. When a failure occurs, an operator can issue a VARY ACQ operator command to acquire these resources without disrupting sessions.

If more than one VTAM host will activate the NCP, you can code a PCCU definition statement for each host. Each PCCU definition statement describes the functions performed by the host that it is associated with. For example, if you want to assign a second host to back up a communication management configuration (CMC) host, code a PCCU definition statement for the backup host. The definition statement is not used until a failure occurs, at which time the backup host activates the NCP.

Note: For an SSCP to successfully take over for another SSCP that has issued a DR MOVE, the takeover SSCP must also support DR MOVE. In other words, the takeover SSCP must support the dynamic reconfiguration MOVE facility. More importantly, the takeover SSCP must be configured.

Problem Determination in a CMC Configuration

If you configure your network to use a data host, you should be aware of the following problem determination restrictions associated with this type of attachment to the NCP:

- The VTAM I/O trace must be used to verify the transfer of data to and from VTAM over the channel link station. The VTAM internal trace can be used if necessary. The line trace function cannot be used even though the NCP is defined on a LINE definition statement.
- The link level 2 test to the NCP is not supported.
- Any function requiring an SSCP-PU session from the data host cannot be supported (for example, NCP storage display; line trace; link level 2 test for resources attached to the NCP; or buffer, TG, or SIT traces for any channel-attached cross-domain NCP).

Dynamic Configuration of Channel-Attached Devices (MVS)

This section describes the following tasks necessary to dynamically configure channel-attached devices:

- Installation and preparation
- Defining your configuration
- Building resource definitions
- Using the default naming convention
- Customizing the command lists.

Installation and Preparation

Information on preparing VTAM to support dynamic configuration of channel-attached devices is detailed in “Data Sets for Dynamic I/O Configuration” on page 27.

To dynamically configure your channel-attached devices, use the NetView program or another product providing a similar interface and services. If you are using NetView, do the following to prepare it for dynamic I/O configuration:

1. Modify the message automation table so that when MVS issues message IOS502I, NetView can start the IOS502I command list. The following sample addition to the message automation table logs messages issued by the ISTDEFIN command list, starts the IOS502I command list, routes the message to one operator and logs it:

```
IF MSGID="ISTDEFIN" THEN
  DISPLAY (N)
  NETLOG(Y);
IF MSGID="IOS502I" THEN
  EXEC (CMD("IOS502I") ROUTE (ONE OPER1))
  DISPLAY (N)
  NETLOG(Y);
```

2. Modify the NetView start procedure to point to the sample library (VTAM.SAMPLIB) where the IOS502I and ISTDEFIN command lists reside or move the command lists from the sample library to the library where your other command lists are stored.
3. Activate the message automation table that you changed using the NetView command AUTOMSG MEMBER=tablename.

Defining Your Configuration

Use the hardware configuration definition (HCD) to interactively define your I/O configuration changes. Use of the HCD is described more fully in the MVS library.

To attach a device to the system:

1. Physically attach the device.
2. Define the device to MVS using HCD.
3. Activate the new configuration using the MVS command ACTIVATE IODF.

MVS can then issue message IOS502I, which causes NetView to start the IOS502I command list. The IOS502I command list calls the ISTDEFIN command list for each device added to the system.

Note: Message IOS502I does not display on the MVS console nor on the NetView operator's console unless requested using the NetView ASSIGN command.

Building Resource Definitions

The ISTDEFIN command list obtains descriptions about the new devices using the ISTDINFO function module. The device-specific information that is returned by the ISTDINFO function module varies according to the device.

The ISTDEFIN command list builds channel-attachment or local SNA major nodes (PU type 2, 2.1, and 4) for the devices.

The ISTDEFIN command uses IBM-supplied defaults to build the definition statements. The following table shows the devices and the definitions that the ISTDEFIN command builds.

Table 16. Definitions for Dynamically Configured Devices

Device	Definition
3705	channel-attachment major node
3720, 3725	Channel-attachment major node
3745 (type 6 adapter)	Channel-attachment major node
3745 (type 7 adapter with NCP loaded)	Channel-attachment or Local SNA major node
3274	Local SNA major node
3174	Local SNA major node

For a 3745 device that is self-describing (3745 type 7 adapter), the ISTDEFIN command builds either a channel-attachment major node or a local SNA major node to define the device. The definition built is determined by the NCP generation already residing in the 3745. For example, an NCP level that does not support a type 7 adapter causes the ISTDEFIN command to build a channel-attachment major node. Otherwise, the major node built depends on whether the channel is defined as a type 2 (Local SNA major node) or type 4 (channel-attachment major node) in the NCP.

Information needed for defining the major nodes that is not provided by the ISTDINFO function module is defined in templates in the ISTDEFIN command list.

Notes:

1. If the device is self-describing, then an NCP being defined must already be loaded.
2. The VTAM using dynamic I/O configuration can only be a data host to an NCP. VTAM does not alter existing PATH definition statements to create a route to the NCP. Do this before you try to dynamically add any devices. Otherwise, the activation of the channel-attachment will fail.
3. A local SNA major node can be dynamically created to represent a type 2.1 channel-attached to an NCP. However, no CDRSC definitions for the independent LUs defined under the type 2.1 channel are dynamically created. Also, default adjacent link station lists belonging to the independent LUs are not updated with the dynamically-created PU name.

Using the Default Naming Convention

The IBM-supplied command lists generate names for resource definitions using either an IBM-supplied or user-written naming convention. Names are generated in the following manner:

1. The first character is N, identifying the name as being generated by VTAM for dynamic I/O configuration.
2. The second character is the subarea number expressed as a single character (0–9 or A–Z). For example, a communication controller in subarea 6 has a second character of 6. A communication controller in subarea 35 has a second character of Z. A communication controller in subarea 36 has a second character of 0, because subarea numbering begins with 1, not 0. The maximum number of subareas that can be defined using the IBM-supplied naming convention is 36.

3. The next four characters are the 4-digit device address, represented in hexadecimal.
4. In the definition of a single device, a number of resources must be named. The last characters depend on the resource being named:
 - L** LINE
 - G** GROUP
 - D** Data-set name
 - xx** 2-digit hexadecimal local address for the logical unit attached to the cluster controller (3x74)

For example, the name generated for the definition of a device at X'123' in subarea 6 is named N60123D. The LINE name for the device is N60123L.

Customizing the Command Lists

You can modify the IOS502I and ISTDEFIN command lists to match your environment. The command lists are coded in REXX, and the subroutines and functions are documented in the code. For a description of the methods for calling the command lists and the information returned, see Appendix F, "Command Lists: Dynamic Configuration of Channel-Attached Devices" on page 673.

Modifying the Resource Definition Defaults

Information needed for defining the major nodes that is not provided by the ISTDINFO function module is defined in templates in the ISTDEFIN command list. The templates contain information on logon modes, interpret tables, resource names, the number of logical units (SNA cluster controllers only), and other information.

Some information, such as the number of buffers VTAM should allocate for receiving data from the NCP (provided on the MAXBFRU operand of the HOST definition statement), is not always available. You can modify the defaults in the templates to override the IBM-supplied defaults. Edit the ISTDEFIN command list, which shows you the templates with which you can modify the values in the REXX statements.

Note: When logical units are being defined, if the physical unit presents a *logical_terminals* string, only the first LU statement in the template is used.

Modifying the Naming Convention

You can change the naming convention by modifying the ISTDEFIN command list. A set of directions is provided in the ISTDEFIN command list for supplying a *seed* value to control the values for the first two characters of the device name. For example, if you want to have your device names begin with MY, change the following in the ISTDEFIN command list:

```
seed='MY'                /* user seed value */
```

Thus, the name generated for the definition of a device at X'123' is named MY0123D.

If you want to use the IBM defaults, do not change the *seed* value.

ISTDEFIN replaces the major node if the same name already exists in USER1.AUTO.VTAMLST, so the definition contains the latest device information.

Supplying Your Own Naming Convention

If you want to use your own naming convention, change the logic of the ISTDEFIN command list. The functions and subroutines that obtain the system parameters used to build the definitions (such as device number and subarea number) are provided and documented in the command list. You can use these functions to modify the ISTDEFIN command list to use your own naming convention.

Writing Your Own Command Lists

The ISTDEFIN and IOS502I command lists are designed to work together. However, you can call these command lists with your own functions and routines.

For example, you can write a command list that is invoked when NetView or VTAM start-up completes. This command list can determine all VTAM-supported devices not currently used by VTAM. This command list can check all possible devices and then call the ISTDEFIN command list to build the definitions and activate the devices.

Dynamic Reconfiguration and Change of Operands

This section contains the following subsections:

- Dynamic Reconfiguration and Dynamic Change Requirements
- Using the VARY ACT, UPDATE Technique
- Using the VARY DRDS Technique
- Using the MODIFY DR Technique
- Dynamic Reconfiguration of Independent LUs
- Dynamic Reconfiguration of Frame Relay Resources
- Implicit Dynamic Reconfiguration
- Dynamic Reconfiguration Examples.

Dynamic Reconfiguration and Dynamic Change Requirements

Table 17 shows which techniques of dynamic reconfiguration are valid for a particular operation on a resource in a particular major node.

Table 17 (Page 1 of 2). Dynamic Reconfiguration Operations for Valid Major Nodes

Operation/Resource	NCP	Channel-attachment or Packet	Local SNA	Switched
Add LU	2, IMP	2	2	–
Delete LU	3	3	3	–
Move LU	1	1	1	–
Change LU operands	1	1	1	1
Add PU	2, IMP	–	–	–
Delete PU	3	–	–	–
Move PU	3	–	–	–

Table 17 (Page 2 of 2). Dynamic Reconfiguration Operations for Valid Major Nodes

Operation/Resource	NCP	Channel-attachment or Packet	Local SNA	Switched
Change PU operands	1	–	–	1

Legend:

- 1 V ACT,UPDATE only
- 2 Either V DRDS or V ACT,UPDATE
- 3 Either MODIFY DR, V DRDS, or V ACT,UPDATE
- IMP Implicit dynamic reconfiguration

Dynamic reconfiguration is not supported on a line that has type 4 or 5 physical units on it (with the exception of type 2.1 PUs and independent LUs on NCP ESCON channels).

VTAM does not redrive changed PU ADDRs or LU LOCADDRs during the initial activation of an NCP (including activation of an NCP during a WARM start), even when VARY ACT,UPDATE=ALL is used. However, if the PU or LU was also moved, VTAM will redrive.

VTAM support for dynamic reconfiguration requires that the PUDRPOOL and LUDRPOOL definition statements be coded at NCP generation. VTAM requests that the NCP assign a PU or LU address to a resource specified by the host. The addresses come from the address pools generated by the LUDRPOOL or PUDRPOOL definition statements. Be sure that LUDRPOOL and PUDRPOOL are large enough to accommodate all of the resources that you plan to add to your network using dynamic reconfiguration.

Similarly, be sure that the MAXPU keyword (on the LINE definition statement) and the MAXLU keyword (on the PU definition statement) are large enough to accommodate all of the resources that you plan to add using dynamic reconfiguration. MAXPU limits the number of PUs on a line, and MAXLU limits the number of LUs under a PU.

In addition, when a PU is added to a line, NCP adds the PU at the end of its service order table. If you want to have the PU added multiple times in the service order table, you must regenerate and reload the NCP.

For more information on PUDRPOOL, LUDRPOOL, MAXPU, MAXLU, and the service order table, see the *NCP, SSP, and EP Resource Definition Reference* and the *NCP, SSP, and EP Resource Definition Guide*.

Using the VARY ACT,UPDATE Technique

You can use the VARY ACT,UPDATE technique to dynamically change certain operand values on PU and LU definition statements without deactivating the major node. Only the PUs and LUs that you are changing must be inactive. To change the values, edit the existing definitions in the VTAMLST file and then issue a VARY ACT,UPDATE=ALL command.

Notes:

1. PUTYPE cannot be changed, and you cannot add SDDL support to a PU.
2. You can convert an APPN PU to a LEN PU, or a LEN PU to an APPN PU, by changing the value of the CONNTYPE operand. However, you cannot convert an APPN PU to a LEN PU if either of the following is true:
 - You coded the same NETID/CPNAME value on multiple PUs.
 - You coded an ADJCP definition statement for a PU to represent a partner CP and you coded the same NETID/CPNAME on that PU.

Some VTAM-only operands can be changed when the resource is inactive. All other operands can be changed only by deleting the resource and adding the resource. For more information on which operands support dynamic change, see “Dynamic Change of Operands” under the local SNA, NCP, and switched major nodes in the *VTAM Resource Definition Reference*.

To use the VARY ACT,UPDATE technique for dynamic reconfiguration, change the VTAMLST source file for a major node and issue a VARY ACT,UPDATE=IMPLICIT command. The command causes VTAM to update the actual configuration to match the new definition in the VTAMLST source file. The major node can be active for ADD or ALL. Only the individual resources being added, deleted, or moved must be inactive.

VARY ACT,UPDATE=IMPLICIT (the default) has the same effect as implicit dynamic reconfiguration. For more information on implicit dynamic reconfiguration, see “Implicit Dynamic Reconfiguration” on page 300. For more information on the VARY ACT command, see “VARY ACT Command” in *VTAM Operation*.

With VARY ACT,UPDATE=ADDIALL, dynamic reconfiguration changes stay in effect regardless of VTAM or NCP recycles (halt and restart) and upon takeover of an NCP. You need not redrive dynamic reconfiguration changes with VARY DRDS or MODIFY DR commands. It is recommended that you **not** make changes to VTAMLST files during a VTAM or NCP recycle. Instead, make the changes dynamically while VTAM and NCP are running. To prepare for takeover of an NCP, distribute the new VTAMLST file to the backup host and issue the same VARY ACT,UPDATE=ADDIALL command that you issued at the primary host.

With VARY ACT,UPDATE=ADDIALL, you can delay the NCP generation for a longer time than you can with other methods of dynamic reconfiguration. The changes you have made in VTAMLST definition files cause more network flows between VTAM and NCP on a recycle, but you need not regenerate the NCP until performance during recycles becomes unacceptably slow. Regenerating and reloading the NCP eliminates the extra flows.

If you use the NCP/EP definition facility (NDF) to generate some definitions, make your changes to the NDF source input, run that source through NDF, and place the resulting NEWDEFN file into a VTAMLST member. Save the load module for future use. If you update the NEWDEFN file directly, your changes are lost the next time NDF is run.

Sifting of operands does take place on subordinate resources that have been added or moved by the VARY ACT,UPDATE method of dynamic reconfiguration.

Network management programs that rely on VTAMLST definitions for the network configuration have current information available to them about the configuration.

Note: If you use:

- The VARY LOGON command to establish an automatic logon specification that is not coded on the LOGAPPL operand of a definition statement
- The VARY NOLOGON command to delete an automatic logon specification that is coded on the LOGAPPL operand of a definition statement
- The MODIFY DEFAULTS, DLOGMOD command to change the value of the DLOGMOD operand of a definition statement
- The MODIFY TABLE command to change VTAM table associations,

the VARY ACT,UPDATE=ALL command might override these commands. For example, the LOGAPPL operand in the VTAMLST definition overrides the VARY LOGON and VARY NOLOGON commands. Even if no LOGAPPL is coded in the VTAMLST definition, a null value for LOGAPPL overrides the VARY LOGON value. To avoid having the VARY LOGON command overridden, code the LOGAPPL definition statement with the desired value. To avoid having the VARY NOLOGON command overridden, delete the LOGAPPL operand from the definition statement.

Coding Dynamic Reconfiguration Changes

To add resources to the network, code the new definition statements in the desired location in the VTAMLST file for the major node. To delete resources, delete the appropriate definition statements from the VTAMLST file. To move resources, move the appropriate definition statements from one location to another location in the VTAMLST file.

You can move a physical unit (and its associated logical units) from a line under one major node to a line under a different major node under the same VTAM. To do this, delete the resources from one major node and add them to the other major node. You can do the deletions with any method of dynamic reconfiguration. Do the additions in the VTAMLST file.

Some move or add operations can depend upon a delete operation completing first. For example, you could exchange resources between two major nodes at the same time (that is, move resources from the first major node to the second major node and also move resources from the second major node to the first). Or, you might delete a PU from a line and add a PU to the line with the same ADDR value as the deleted PU.

You can resolve this problem in either of these ways:

- First do the deletions with any method of dynamic reconfiguration, then do the additions by modifying VTAMLST and issuing the VARY ACT,UPDATE=ADD command.
- Process the source definitions twice (issue the same VARY ACT command again). The deletions will take place on the first run of the command, and the additions that failed on the first run will take place on the second run.

Errors in the VTAMLST Definition File

If you attempt dynamic reconfiguration and the VTAMLST file contains errors, you might not get the results that you expected. Errors in BUILD or PCCU statements or errors in required operands can cause the VARY ACT command to fail, in which case no dynamic reconfiguration takes place. Errors in definition statements can cause unplanned dynamic reconfiguration changes or unplanned dynamic changes in operand values. Depending on the severity of the error in the definition file, VTAM may skip over resources that have incorrect definitions and process the rest of the definition file. VTAM deletes resources that are skipped unless they are active.

Note: It is recommended that you run the NCP/EP definition facility (NDF) or issue the VARY ACT,SCOPE=SYNTAX command to check for definition errors before you attempt dynamic reconfiguration changes.

Using the VARY DRDS Technique

The VARY DRDS command activates a dynamic reconfiguration (DR) file. To use the VARY DRDS command, first define the appropriate dynamic reconfiguration files. When the files have been defined, you can issue the VARY DRDS command to implement the changes. You can code a dynamic reconfiguration file to:

- Move a physical unit from one nonswitched line to another within the same NCP
- Add physical units to nonswitched lines, or add logical units to physical units under nonswitched lines
- Change the link station address of physical units under nonswitched lines
- Delete physical units from nonswitched lines and delete logical units from physical units under nonswitched lines
- Delete NCP frame relay PUs.

The NCP to be reconfigured must be active, and the resource to be deleted or moved must be inactive. Also, the NCP cannot recognize move or add requests unless the physical unit was active at some time before inactivating it to issue a VARY DRDS command on it.

Dynamic reconfiguration with a DR file should be used only as a temporary method; a new NCP should be generated to reflect the changes as soon as time permits. The NCP should be reloaded to reflect the permanent changes.

If you are coding a dynamic reconfiguration file:

1. The first definition statement in the file must be VBUILD TYPE=DR. Include appropriate ADD, DELETE, and MOVE statements for the changes you want to make. For each statement, provide VTAM definitions for the resources you want to change.

Code the proper value (LOCAL, CA, or PACKET) for the DRTYPE operand on the ADD and DELETE definition statements to indicate that the dynamic reconfiguration operation is for a connection through the local peripheral node. The default (DRTYPE=NCP) does not work in this case.

2. File the dynamic reconfiguration file in the VTAM definition library.
3. Make sure that you have deactivated all resources that must be inactive.

4. Issue the VARY DRDS command to invoke dynamic reconfiguration using the named dynamic reconfiguration file.
5. Issue the VARY ACT command to activate both newly added resources and resources that you deactivated for the reconfiguration.

To automatically activate a resource added by dynamic reconfiguration, the status of the higher-level node should be ACTIVE, and ISTATUS=ACTIVE should be specified on the newly added resource.

To automatically activate a moved physical unit and its logical units that have ISTATUS=ACTIVE coded, specify ACTIVATE=YES on the PU definition statement following a MOVE statement.

6. Check the results of the dynamic reconfiguration by issuing a DISPLAY command for the affected resources.

MVS, VSE If the NODELST start option is in effect when you perform a dynamic reconfiguration, VTAM records the name of each dynamic reconfiguration file in the NODELST file. When you restart VTAM after a halt or a failure, you can automatically reapply the changes in the dynamic reconfiguration files by specifying the name of the NODELST file on the CONFIG start option.

If a physical unit was dynamically moved before VTAM failed, it may not be moved during a WARM start. The activation of resources precedes the processing of dynamic reconfiguration files during warm start. Because dynamic reconfiguration depends upon resources being inactive, dynamic reconfiguration move operations are not performed. Inactivate the PU and issue the VARY DRDS command again.

Note: VARY ACT,UPDATE enables resources to be dynamically reconfigured and activated on warm start.

Adding Resources

For an ADD operation, code the PU and LU definition statements as you would when defining these resources. If you do not code the MAXDATA operand on the PU definition statement, a value of 261 bytes is used for a PU type 1 and a value of 265 bytes is used for a PU type 2. Code the operands on the PU and LU definition statements in the same format as they appear in the NCP major node. For example, ensure that PASSLIM and MAXOUT are coded correctly. For the coding format of NCP operands, refer to the *NCP, SSP, and EP Resource Definition Reference*.

In a DR file, explicitly code all operands from the GROUP, LINE, or PU to which you are adding the PU or LU. VTAM does not process most NCP-only operands and might issue a warning message, but the GP3174 and RETRIES=(,t,n) NCP operands can be used when adding PUs on a nonswitched SDLC link.

Sifting takes place within the hierarchy of minor nodes being added dynamically. Values for operands coded in the original hierarchy above the hierarchy being added with the DRDS deck do not sift down to the added resources. Therefore, if you want the values coded in the original hierarchy, code them explicitly in the DR file for the resources you are adding.

The most efficient way to add a physical unit and its associated logical units with a DR file is to specify them on the same ADD statement. It is not necessary to use separate ADD statements for the physical unit and each logical unit. Figure 73 on page 301 shows how to code a DR file to add PUs and LUs.

Deleting Resources

For a DELETE operation, code only the PU or LU names in the DR file. You do not have to code individual PU or LU operands. When you delete a PU, you also delete its associated LUs. Do not code the associated LU definition statements. Figure 74 on page 302 shows how to code a DR file to delete PUs and LUs.

Moving Resources

For a MOVE operation using a DR file, code only the names of the PUs you want to move. Do not code associated LUs. VTAM automatically includes the associated LUs in the move. For each PU, you can optionally specify a new SDLC link station address and whether you want the PU and its LUs automatically activated after the move. Figure 75 on page 303 shows how to code a DR file to move PUs and LUs.

To change the SDLC link station address for a physical unit on a certain line, use the MOVE statement and specify the new address for the ADDR operand, keeping the FROM line and the TO line the same.

The MOVE operation via a DR file is only for PUs (and their associated LUs). To move an individual LU to another PU, use a DELETE and then a subsequent ADD. (For efficient storage management, code all DELETES before their related ADDs.)

Note: You can move a physical unit only under the same NCP major node. The physical unit cannot be moved from a line attached on one NCP to a line associated with another NCP.

Using the MODIFY DR Technique

The MODIFY DR command allows you to move a PU (and its LUs) from one line to another line, delete a PU (and its LUs) from a line, or delete an LU from a PU. Figure 74 on page 302 shows how to delete PUs and LUs using the MODIFY DR command. Figure 75 on page 303 shows how to move a PU and its LUs using the MODIFY DR command. You can change the SDLC address for a PU by requesting a move in which the new line and old line are the same, but the address is different.

The NCP to be reconfigured must be active, and the resource to be deleted or moved must be inactive.

Dynamic reconfiguration with the MODIFY DR command should be used only as a temporary method; a new NCP should be generated to reflect the changes as soon as time permits. The NCP should be reloaded to reflect the permanent changes.

MVS,VSE Dynamic reconfiguration changes made by MODIFY DR are not recorded in NODELST. Therefore, they are not automatically applied and can cause errors because they are not used during a warm start.

Dynamic Reconfiguration of Independent LUs

You can dynamically add or move independent LUs with the VARY DRDS command or the VARY ACT command with the UPDATE operand. You can also dynamically change operand values in an independent LU with the VARY ACT,UPDATE=ALL command. For a dynamic reconfiguration add or move, or for dynamically changing operand values, the processing takes place according to Table 18 on page 298. For further information, see "Dynamic Reconfiguration and

Dynamic Change” and “Rules for Multiple Definition of Resources” in the *VTAM Resource Definition Reference*.

Table 18. Rules for Multiple Definition of Resources

Existing resource	Input resource	Major node after integration	Notes
Dynamic CDRSC	DR-defined independent LU	ISTPDILU	1, 2
Predefined independent LU	DR-defined independent LU	ISTPDILU	1, 2
Predefined CDRSC	DR-defined independent LU	ISTPDILU	3, 3

Notes:

1. If the existing resource has a known real NETID, it must match the NETID of the input resource. If the real NETID of the existing resource is not known, the NETID is inherited from the input resource.
2. All operands from the input resource apply to the existing resource.
3. Only LU operands from the input resource apply to the existing resource.

Dynamic Reconfiguration of Frame Relay Resources

Frame relay PUs, which are type 1 PUs, are used in FRSESET (frame relay switching equipment set) definition statements to create routing paths within an NCP. Each FRSESET definition statement can contain from two to four frame relay PUs, which define the primary routing partners and their optional backups. In the following example, PU A and PU B are the primary partners, while PU C and PU D are the backups. PU C is the backup for PU A, and PU D is the backup for PU B.

```
frse3a3 FRSESET SUBPORTS=(A,B,C,D)
```

If one of the primary partners is inactivated, routing can continue between the backup for the inactive primary PU and the remaining primary PU. Backup to backup routing is not permitted.

The VARY DRDS, MODIFY DR, and VARY ACT,UPDATE=ADDIALL commands can be used to dynamically reconfigure frame relay PUs and FRSESET definition statements. To delete a PU from a FRSESET, use VARY DRDS, MODIFY DR, or VARY ACT,UPDATE=ALL. To add a PU to a FRSESET, use VARY ACT,UPDATE=ADDIALL. Moving frame relay PUs is not allowed, but a move can be accomplished by deleting a PU and then adding it back in the desired position.

The following rules apply when using dynamic reconfiguration for frame relay resources:

- Each FRSESET definition statement must have a domain-unique name.
- The LMI (local management interface) PU must be statically defined as the first PU under a frame relay line. The LMI PU cannot be used in a FRSESET and cannot be added (or deleted) dynamically.
- Each PU in a FRSESET must be on a different frame relay line.
- A PU cannot be concurrently defined in multiple FRSESET definition statements.

- Use UPDATE=ALL for initial activation. If VTAMLST is modified prior to initial activation, using UPDATE=IMPLICIT can cause NCP and VTAM to have inconsistent definitions. If this occurs, issue a VARY ACT,UPDATE=ALL command to correct the problem.
- To modify a statically defined FRSESET, all PUs in the FRSESET must first be deleted.

Note: Statically defined PUs that are deleted and then added back (for example, when adding backups to a statically defined FRSESET) are regarded as dynamic PUs.

- A FRSESET cannot contain both static and dynamic PUs.
- To delete a PU from a FRSESET, you must first inactivate the PU. Otherwise, the deletion fails and the PU is still present.
- Do not delete either of the primary PUs from a dynamically defined FRSESET. Otherwise, remaining PUs in the FRSESET may be inactivated, and the FRSESET cannot be used until a primary PU is added.
- PUs must be deleted from the frame relay line and the FRSESET simultaneously.
- When adding PUs to a FRSESET, PUs must be added to the frame relay line and the FRSESET simultaneously.

If these rules are not followed, you may receive messages IST322I, IST323I, and IST1350I, and you can do the following to correct the problem:

- If the FRSESET was statically defined, delete all PUs and the FRSESET, correct the error, and then add back the PUs and the FRSESET.
- If the FRSESET was dynamically defined, only the PU or FRSESET in error must be corrected.

If any PU in a dynamic FRSESET is inactivated, all other active or contact pending PUs in the FRSESET that have no potential routing partner become inoperative. The inoperative PUs go through recovery processing and are again active, although not currently usable for routing since they have no partner.

Data traffic flowing over the inactivated PU could be momentarily halted while the switch to the backup PU is performed. If the backup is not available, data traffic ceases. Data traffic also ends if the other primary PU is not available; backup to backup routing is not permitted.

If routing through a backup PU and the primary PU becomes available, routing automatically switches back to the primary PU.

For further information on frame relay PUs, see “DISPLAY ID Command,” “MODIFY DR Command,” “VARY ACT Command,” or “VARY DRDS Command” in *VTAM Operation*, and “FRSESET Definition Statement” in the *VTAM Resource Definition Reference*. Information can also be found in the *NCP, SSP, and EP Resource Definition Reference*, and the *NCP, SSP, and EP Resource Definition Guide*.

Implicit Dynamic Reconfiguration

Implicit dynamic reconfiguration allows you to reconfigure an NCP by changing VTAM's definition of the NCP and then reactivating the NCP. This procedure is disruptive in that deactivation of an NCP causes all existing sessions to be terminated. As soon as possible, you should regenerate and reload the NCP to reflect the changes. To use implicit dynamic reconfiguration:

1. Modify the existing NCP generation definition and include new definition statements for the PUs and LUs you want to add. You can only add PUs and LUs. You cannot delete or move them.
2. Issue a VARY INACT command to deactivate the NCP.
3. Issue a VARY ACT,UPDATE=IMPLICIT (the default) command to reactivate the NCP, using the updated generation definition.

For an example of adding PUs and LUs using implicit dynamic reconfiguration, see Figure 76 on page 304.

Note: The UPDATE=IMPLICIT operand on the VARY ACT command provides the same support that was available for pre-V4R1 releases.

Dynamic Reconfiguration Examples

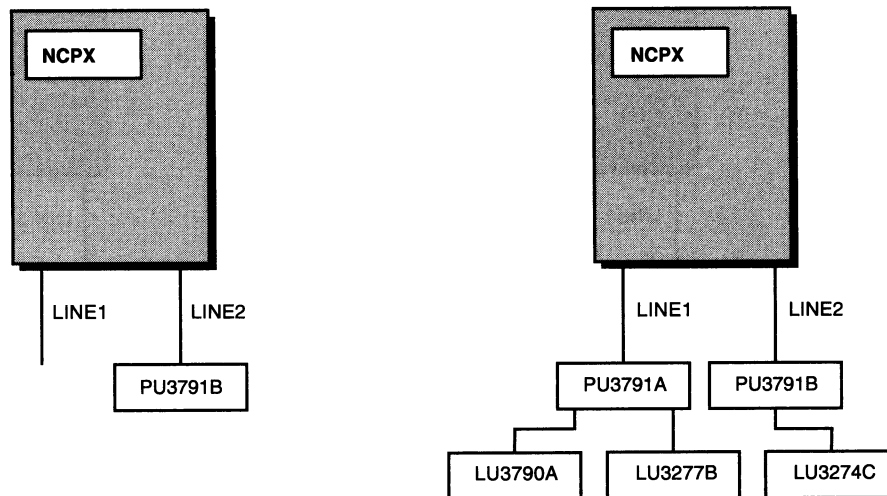
Figure 73 shows how to perform a dynamic reconfiguration ADD operation. You can modify the VTAMLST source for the desired major node and then use the VARY ACT,UPDATE=ALLIADD command, including the member name of the data set containing the updated definition. Changes made in this way are permanent changes.

Alternatively, you can make temporary changes by creating a DR file and issuing the VARY DRDS command. These changes do not remain in effect when the NCP is regenerated and reloaded.

The MODIFY DR command cannot be used to add resources.

Original VTAMLST definition deck:

```
LINE1      LINE (Normally coded operands)
LINE2      LINE
PU3791B    PU
```



Using VARY ACT, UPDATE

Modified VTAMLST definition deck:

```
LINE1      LINE (Normally coded operands)
PU3791A    PU
LU3790A    LU
LU3277B    LU
LINE2      LINE
PU3791B    PU
LU3274C    LU
```

Operator command: VARY NET,ACT,UPDATE=ADDIALL,ID=member name

Using VARY DRDS

DR File:

```
VBUILD TYPE = DR
ADD TO = LINE1
PU3791A PU (Normally coded operands)
LU3790A LU (Normally coded operands)
LU3277B LU (Normally coded operands)
ADD TO=PU3791B
LU3274C LU (Normally coded operands)
```

Operator command: VARY NET,DRDS,ID=dr filename

Figure 73. DR ADD Operation

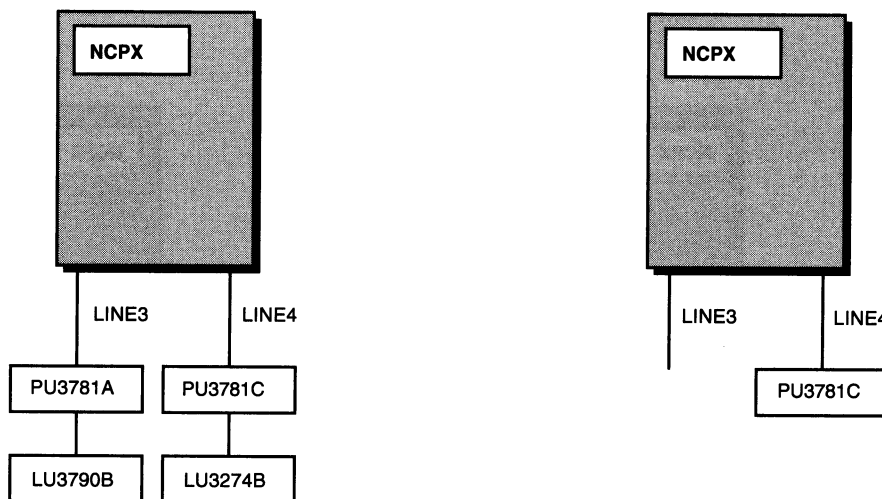
Figure 74 shows how to perform a dynamic reconfiguration DELETE operation.

You can remove the PU and LUs from the VTAMLST definition and issue a VARY ACT,UPDATE=ALL command. Changes made in this way are permanent changes.

Alternatively, you can make temporary changes by creating a DR file and issuing the VARY DRDS command, or by using MODIFY DR commands. These changes do not remain in effect when the NCP is regenerated and reloaded.

Original VTAMLST definition deck:

```
LINE3    LINE (Normally coded operands)
PU3781A  PU
LU3790B  LU
LINE4    LINE
PU3781C  PU
LU3274B  LU
```



Using VARY ACT, UPDATE

Modified VTAMLST definition deck:

```
LINE3    LINE (Normally coded operands)
LINE4    LINE
PU3781C  PU
```

Operator command: VARY NET,ACT,UPDATE=ALL,ID=member name

Using VARY DRDS

DR file:

```
VBUILD TYPE=DR
DELETE FROM=LINE3
PU3791A PU (No operands required)
DELETE FROM=PU3781C
LU3274B LU (No operands required)
```

Operator command: VARY NET,DRDS,ID=dr filename

Using MODIFY DR Operator Commands

```
MODIFY DR,TYPE=DELETE,FROM=LINE3,ID=PU3781A
MODIFY DR,TYPE=DELETE,FROM=PU3781C,ID=LU3274B
```

Figure 74. DR DELETE Operation

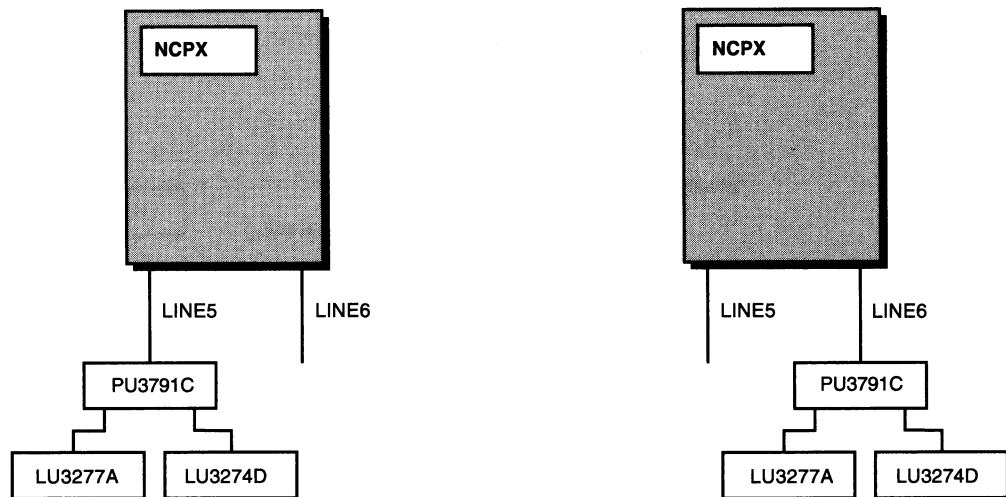
Figure 75 shows how to perform a dynamic reconfiguration MOVE operation.

You can edit the VTAMLST definition to move the PU and LU definitions to the desired locations and issue a VARY ACT,UPDATE=ALL command. Changes made in this way are permanent changes.

Alternatively, you can make temporary changes by creating a DR file and issuing the VARY DRDS command, or by using MODIFY DR commands. These changes do not remain in effect when the NCP is regenerated and reloaded.

Original VTAMLST definition deck:

```
LINE5      LINE (Normally coded operands)
PU3791C    PU
LU3277A    LU
LU3274D    LU
LINE6      LINE
```



Using VARY ACT, UPDATE

Modified VTAMLST definition deck:

```
LINE5      LINE (Normally coded operands)
LINE6      LINE
PU3791C    PU
LU3277A    LU
LU3274D    LU
```

Operator command: VARY NET,ACT,UPDATE=ALL,ID=member name

Using VARY DRDS

DR file:

```
VBUILD    TYPE=DR
MOVE      TO=LINE6, FROM=LINE5
PU3791C    PU      (No operands required)
```

Operator command: VARY NET,DRDS,ID=dr filename

Using MODIFY DR Operator Command

MODIFY DR,TYPE=MOVE,ID=PU3791C,FROM=LINE5,TO=LINE6

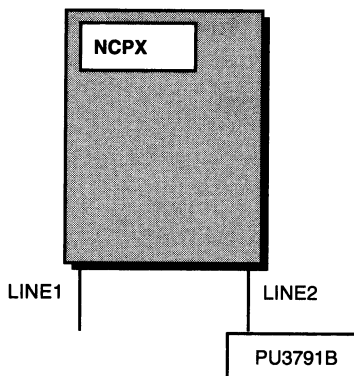
Figure 75. DR MOVE Operation

Figure 76 shows how to add a PU and LUs by implicit dynamic reconfiguration.

Note: IMPLICIT is the default for the UPDATE operand on the VARY ACT command.

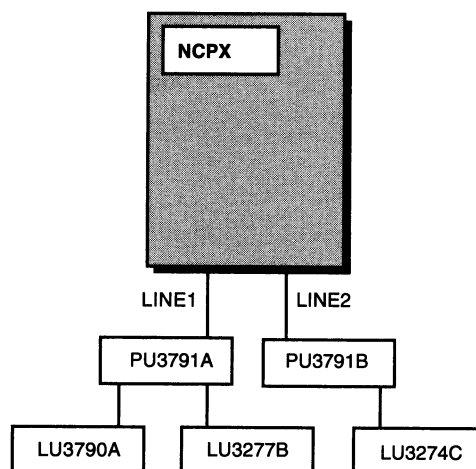
Original NCP Definition Deck:

LINE1	LINE	(Normally coded operands)
LINE2	LINE	
PU3791B	PU	



Modified NCP definition deck:

LINE 1	LINE	(Normally coded operands)
PU3791A	PU	
LU3790A	LU	
LU3277B	LU	
LINE2	LINE	
PU3791B	PU	
LU3274C	LU	



Operator commands:

```
VARY NET,INACT,ID=NCPX
VARY NET,ACT,ID=NCPX,UPDATE=IMPLICIT
```

Figure 76. Sample Implicit Dynamic Reconfiguration

Line Backup through Dynamic Reconfiguration

Dynamic reconfiguration can be used to logically move physical and logical units (the station to be reconnected using switched network backup) to an alternate line. The alternate, physical, switched line is generated in the NCP as a nonswitched line. When a line to a multipoint station fails, the affected station can be deleted from its position as originally generated in the NCP and added to the switched link through dynamic reconfiguration or with the MODIFY DR command. It can then be physically connected by an operator dialing the remote modem. If several stations are attached by a remote “fan out” modem, all the physical units on that modem can be added to the same backup link with the DR ADD command.

The physical units and logical units can have the same names so no changes to application programs are needed.

Figure 77 on page 305 illustrates dynamic reconfiguration without “fan out” modems.

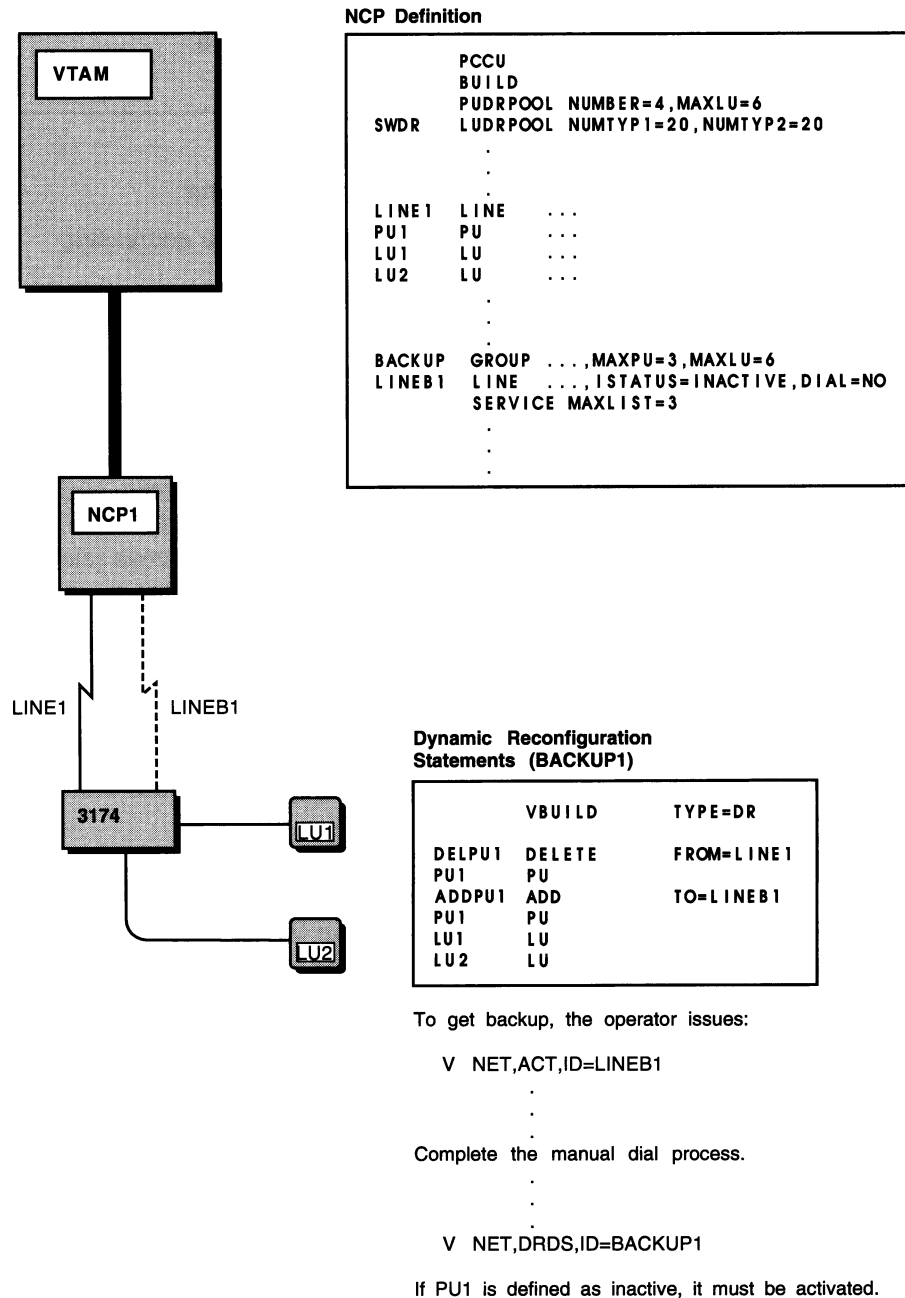


Figure 77. Line Backup through Dynamic Reconfiguration

To define dynamic reconfiguration with both LPDA-1 modems and LPDA-2 DCEs:

1. Generate the NCP to include dynamic reconfiguration.
2. Generate alternate ports in the NCP as nonswitched lines.
 - For LPDA-1 modems, generate LPDA-1 lines.
 - For LPDA-2 DCEs, generate LPDA-2 lines.
3. Create the dynamic reconfiguration definitions and store them in the VTAM definition library. The file should include the following:
 - DELETE statements for the physical units being backed up
 - ADD statements to add these physical units to the backup link.
4. Activate the backup line.

5. Enter a VARY DRDS command to invoke the dynamic reconfiguration. The backup switched line is now ready for use.

Network Routing for Subarea Nodes

This section contains the following subsections:

- How VTAM handles network and subarea addressing
- Virtual route pacing
- Parallel sessions using parallel transmission groups
- Dynamic path update.

How VTAM Handles Network and Subarea Addressing

For VTAM to control routing in a subarea network or within a composite network node, VTAM must know the location of its resources, such as logical units, physical units, and other SSCPs. VTAM uses element addresses, in conjunction with the subarea address, to identify the location of resources (also known as network addressable units). The subarea address indicates in which subarea the resource is located; the element address indicates the unique address within the subarea.

Minor nodes, such as an application program or logical unit, require at least one element address. Some minor nodes, such as local non-SNA devices and application programs that use parallel sessions, require more than one element address. This can increase the number of element addresses that are used in a subarea.

The different types of addressing structures are:

Pre-extended network addressing

Used by releases prior to VTAM V3R1. Enables you to define 255 subareas and 254 elements.

Extended network addressing (ENA)

Used by VTAM V3R1 and subsequent releases prior to VTAM V3R2. ENA supports addressing to 255 subareas and extends element addressing to 32768 elements. Beginning with VTAM V4R1, host element addressing was extended to 65535 elements.

Extended subarea addressing

Used by VTAM V3R2 (with compatibility PTF) and subsequent releases. Extended subarea addressing increased the size of the subarea address to 65535 and the number of explicit routes for each destination subarea to 16. To increase the number of APPN sessions VTAM can support, VTAM V4R2 allows extended host element addressing using the ENHADDR=YES start option. If ENHADDR=YES is specified, high order addresses are assigned for the primary LU when the addresses of both the primary LU and secondary LU are assigned by the same VTAM (not by an NCP). This includes only sessions where both session partners are either a VTAM application or an LU that is attached to VTAM (not to an NCP).

Note: Applications are assigned a low-order primary LU address upon activation. This low-order address may be used in cases where a high-order address could have been used.

You can use the MAXSUBA start option or the MAXSUBA operand in NCP to enable subareas with different addressing structures to communicate. The

MAXSUBA start option specifies the highest subarea value that can be used throughout the network. A MAXSUBA of 63, for example, defines a network with up to 63 subareas and 1024 elements in each subarea. Code the MAXSUBA start option in VTAM's start option list if you want VTAM to communicate with non-extended addressing nodes or if the MAXSUBA operand is coded in an NCP that VTAM communicates with.

Compatibility between Non-Extended and Extended Network Addressing Nodes

Non-extended network addressing nodes can connect to an extended network addressing environment under the following two conditions:

- A compatibility PTF must be installed on all non-extended network addressing nodes that communicate with extended network addressing nodes.
- All nodes that communicate with non-extended addressing nodes must define either the MAXSUBA operand in the start option or the NCP definitions. The value specified for MAXSUBA must be the same at all nodes within the same network.

In addition, the following restrictions apply when extended and non-extended addressing nodes coexist in the same network:

- VTAMs that are non-extended addressing nodes cannot communicate with any logical unit whose element address exceeds the maximum element number defined by the subarea and element split, which is defined by the MAXSUBA operand.
- VTAMs that are non-extended addressing nodes cannot communicate with any node whose subarea is greater than MAXSUBA.

Figure 78 on page 308 illustrates an element address incompatibility.

Although it is not recommended, a network containing non-extended addressing nodes can have subareas where the subarea numbers are greater than the MAXSUBA. In a single-domain network, any NCP whose subarea number exceeds the MAXSUBA address range must be controlled by an ENA VTAM. However, an ENA NCP whose subarea is less than the MAXSUBA address range can be controlled by a pre-ENA VTAM.

Non-ENA NCPs (such as A30NV2) can be adjacent to an NCP whose subarea is greater than the MAXSUBA only if the NCP is used as an intermediate routing node. For example, A240NV4 has a subarea address of 240, which is greater than the MAXSUBA of 63. However, A240NV4 can be adjacent to A30NV2 because A30NV2 is used as an intermediate routing node that transmits data between A240NV4 and host A60M. This function requires that a PTF be applied to A30NV2, the non-extended addressing NCP.

Table 19 on page 308 and Table 20 on page 308 show the adjacent and endpoint node subarea requirements.

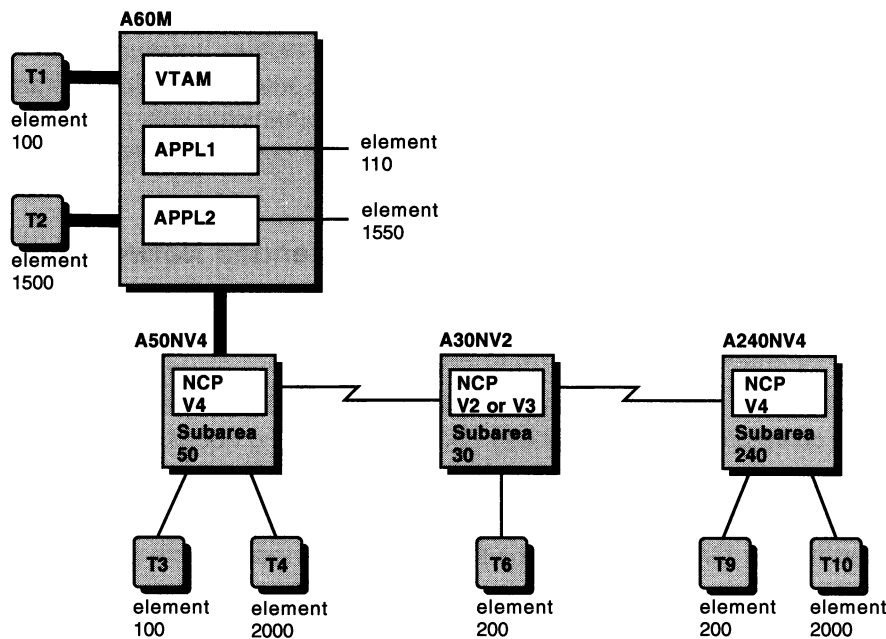


Figure 78. Element and Subarea Address Incompatibility in Single-Domain Environment. The domain contains nodes that use ENA (A60M and A50NV4) and nodes that use non-ENA (A30NV2). All nodes have a MAXSUBA of 63. Although all of the subareas are within the MAXSUBA, some of the element addresses exceed the maximum, which is 1023. The following are the address constraints in this network:

- Terminal T6, attached to link-attached NCP A30NV2, cannot communicate with application program APPL2 because its element address is 1550. However, T6 can communicate with APPL1 or any other application program whose element address is less than 1024.
- If terminal-to-terminal communication is supported in this environment, T6 cannot communicate with terminals T2, T4, and T10 because their element addresses exceed 1023.

Table 19. Adjacent Node Subarea Requirements for Single-Domain Environment

	Pre-ENA VTAM	ENA VTAM Subarea ≤ MAXSUBA	ENA VTAM Subarea > MAXSUBA
Pre-ENA NCP	Yes	Yes	Yes *
ENA NCP Subarea > MAXSUBA	No	Yes	Yes

Note:

- Yes These two nodes can be endpoints.
- No These two nodes cannot be endpoints.
- * Pre-ENA is used as an intermediate routing node only.

Table 20. Endpoint Node Subarea Requirements for Single-Domain Environment

	Pre-ENA VTAM	ENA VTAM Subarea ≤ MAXSUBA	ENA VTAM Subarea > MAXSUBA
Pre-ENA NCP	Yes	Yes	No
ENA NCP Subarea > MAXSUBA	No	Yes	Yes

Note:

- Yes These two nodes can be endpoints.
- No These two nodes cannot be endpoints.

Compatibility between Extended Network and Extended Subarea Addressing Nodes

Non-extended subarea addressing nodes can connect to an extended subarea addressing network, but the connectivity considerations in such networks can be complex. You can simplify migration by installing levels of software that support extended addressing on all subareas in the network and by coding the SALIMIT operand to accommodate planned growth before adding any subareas with a higher SALIMIT value. If you do not do this, some subareas no longer have connectivity to other parts of the network.

To control the maximum number of subareas in a network, the SALIMIT and ERLIMIT operands for VTAM and NCP enable you to specify the maximum subarea size of the network and the maximum number of explicit routes allowed by an NCP.

SALIMIT should, in general, be constant throughout the network. If SALIMIT cannot be constant, you can try different values with no loss of connectivity until you use a subarea address that is higher than the lowest SALIMIT in the network. When moving to a lower SALIMIT value, you should lower all subarea numbers before changing the SALIMIT value.

The following are restrictions in a network that combined extended subarea addressing with other addressing structures:

- Adjacent subareas can support different maximum subarea values and still communicate, provided their actual subarea addresses fall within their common range. For example, a non-extended subarea addressing VTAM can communicate with an extended subarea addressing NCP that supports an SALIMIT value greater than 255 if the NCP's subarea address is not greater than 255.
- A subarea can communicate through a subarea with an address higher than it supports, provided there is an intermediate subarea with an address acceptable to both. For example, a non-extended subarea addressing VTAM could communicate through an NCP with a subarea address of 300. There would, however, have to be an intermediate VTAM or NCP supporting extended subarea addressing with an address no greater than 255 and an SALIMIT of at least 300.
- An SSCP cannot activate any subareas with addresses greater than its addressing capabilities.
- For an extended subarea addressing NCP to successfully send a CONTACT request unit to an adjacent extended subarea addressing NCP with an address greater than 255, the link station must be owned by an extended subarea addressing VTAM.
- To define an explicit route with a number greater than 7, all subareas on the route must support extended subarea addressing and must have ERLIMIT=16 coded on their definition statements.

Virtual Route Pacing

The term virtual route pacing applies to the monitoring and flow control mechanisms that are used for virtual routes. On a network-wide basis, subarea nodes (VTAM and NCP) continuously monitor the amount of traffic in the network. If congestion occurs, these subarea nodes can limit the amount of data they send over the affected virtual routes until the congestion clears. The monitoring of data flow and limiting of data occurs automatically, requiring no action by end users or domain operators.

Activation of a virtual route includes initializing the pacing count. This count, the virtual route's window size, indicates the number of path information units (PIUs) that are allowed to flow on a virtual route before the subarea node receiving the PIUs authorizes the sending of more data. The set of PIUs that can be sent is called a window. This value fluctuates between a minimum and a maximum value depending on network congestion. (Network congestion is determined by how many messages are queued.) You can affect how VTAM does virtual route pacing through the PATH definition statement or the VR pacing window size calculation module. The minimum and maximum values can be specified on the NCP PATH definition statements and the VTAM PATH definition statements that define the virtual routes.

Degrees of Congestion: The network detects two degrees of congestion: minor and severe. Minor congestion is detected when the amount of data queued for a specific transmission priority to a particular transmission group exceeds a user-specified value. Severe congestion is detected when the aggregate amount of data for all transmission priorities exceeds the total user-specified value for a transmission group. At the first, less severe level, the number of PIUs allowed to be sent in a window is decreased by one until congestion clears. At the severe level, the window size is immediately decreased to its minimum value. These two levels can be specified with NCP transmission group flow control threshold parameters. These threshold parameters can be coded on the ER operand on the NCP PATH definition statement.

VTAM reports only severe congestion. This occurs when VTAM is experiencing a buffer shortage.

Local Flow Control and Severe Congestion: Virtual route pacing is supplemented by a local flow control function that limits the amount of data that end users within a subarea can present to the network for transmission. Figure 79 on page 312 illustrates local flow control.

Suppose, for example, that the virtual route between NCP1 and VTAM is congested and that VTAM has been notified of the congestion. If NCP1 does not give VTAM permission to send another window of data, then the virtual route is held. VTAM then keeps the windows of data in IOBUF buffer space until the data can be passed.

If LU1 is an SNA device and session pacing is being used on the LU-LU session, NCP1 withholds the session pacing response going to LU1 until the virtual route is no longer congested. (See Example A in Figure 79 on page 312.) Because virtual route pacing controls only normal-flow requests, LU1 can still send responses to previously received requests as well as expedited flow requests. Because those RUs can also congest the virtual route, NCP1 holds them until the congestion clears.

If LU1 is a non-SNA device (or an SNA device that is not using session pacing on the LU-LU session), NCP1 does not read any input from the device until the virtual route is no longer congested. (See Example C in Figure 79 on page 312.)

When the application program (LU2) in Host1 sends data to a device attached to NCP1, VTAM holds the data in IOBUF buffer pool space until it is informed that the virtual route is no longer congested. After the virtual route is blocked and VTAM holds the windows of data, VTAM queues the application program's SEND request and does not read the data from the user's buffer. (See Example B in Figure 79 on page 312.) The direction of data flow is paced independently. One direction might be congested while the other direction is not. Figure 79 on page 312 shows a virtual route that is congested in both directions.

Note: If VTAM determines that sessions with one of its local devices are contributing to congestion over a route, it shuts off data transmission from the device. Therefore, if there are a number of SNA devices attached to a channel-attached cluster controller, sessions for all the devices are shut down, regardless of whether their sessions are being conducted over the congested route or another route. To avoid this, you might want to specify inbound session pacing for all channel-attached SNA devices so that they are each paced separately. See "Session Pacing" on page 330.

Window Sizing: The IBM-supplied algorithm for window size calculation is designed to work with the route pacing algorithm used in the network. It is appropriate for most installations and configurations. VTAM calculates the minimum and maximum sizes of virtual route pacing windows based on the link protocol and the explicit route length (that is, the number of transmission groups in the explicit route used by the virtual route). While the virtual route is being used to transmit data, adjacent subarea nodes on the route automatically adjust the window sizes within the minimum and maximum limits according to traffic conditions along the route.

However, after tuning VTAM and analyzing traffic patterns and resource capabilities, you might want to choose your own bounds or code a replacement routine that sets the window sizes to different values than the ones supplied by IBM. This exit routine is appropriate for systems where the number of resources could vary considerably from one day to the next. For example, you might find one or more resources whose capacities are not consistently used. This might warrant increasing the window sizes. Decreasing the window sizes is less likely to be useful because network flow control protocols are designed to prevent congestion, and setting window sizes too small could reduce traffic flow considerably.

If you use the Information Management System (IMS), you might want to code a replacement virtual route pacing window size calculation exit routine to reduce the number of virtual route pacing responses that VTAM processes for every IMS transaction. For more information about coding the exit routine for IMS, see "Using the VR Window Size Calculation Exit Routine for IMS (MVS)" in *VTAM Customization*.

You can also specify default minimum and maximum window sizes on the VRPWSnn operand of the PATH definition statement when defining a virtual route. This is the simplest way to choose the default and is sufficient for most needs. If you do not specify VRPWSnn, VTAM uses the default algorithm described previously.

The virtual route pacing window size calculation exit routine can be used in addition to VRPWSnn operands (coded on the VTAM and the NCP PATH definition state-

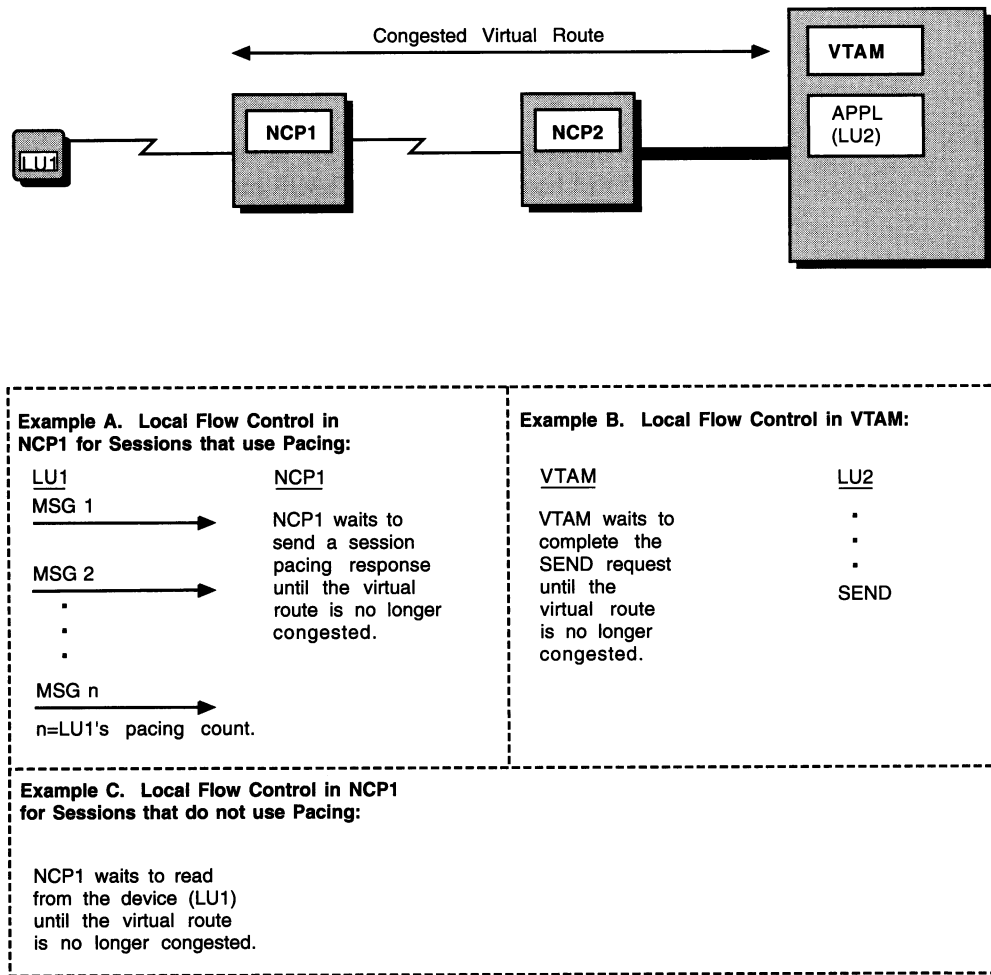


Figure 79. Local Flow Control

ments) to alter the VTAM virtual route pacing window sizes. This exit routine is used to specify the bounds for virtual route pacing windows. A virtual route pacing window represents the quantity of path information units (PIUs) that can be transmitted on a virtual route before a virtual route pacing response is received. This response indicates that the virtual route receiver is ready to receive more PIUs on the route. The exit routine is called when a virtual route is activated. It returns the minimum and maximum values for the window of the virtual route.

Parallel Sessions Using Parallel Transmission Groups

If the application program supports parallel sessions, each session can use a different class-of-service entry and can map to a route that uses a different channel path.

For example, in Figure 80 on page 313, because the type 2.1 peripheral node is connected to an NCP with parallel channel link transmission groups (TG1 and TG2), the independent logical unit (APPL21) can establish parallel sessions with CICS in the host, and each session can use a different route. However, each session must use a different class-of-service entry (such as COS1 and COS2). These different class-of-service entries map to different virtual routes (such as VR1 and VR2) using the same or different transmission priorities (in this case, the same, TP1). As shown in Figure 80 on page 313, the different virtual routes map to the explicit routes for each session.

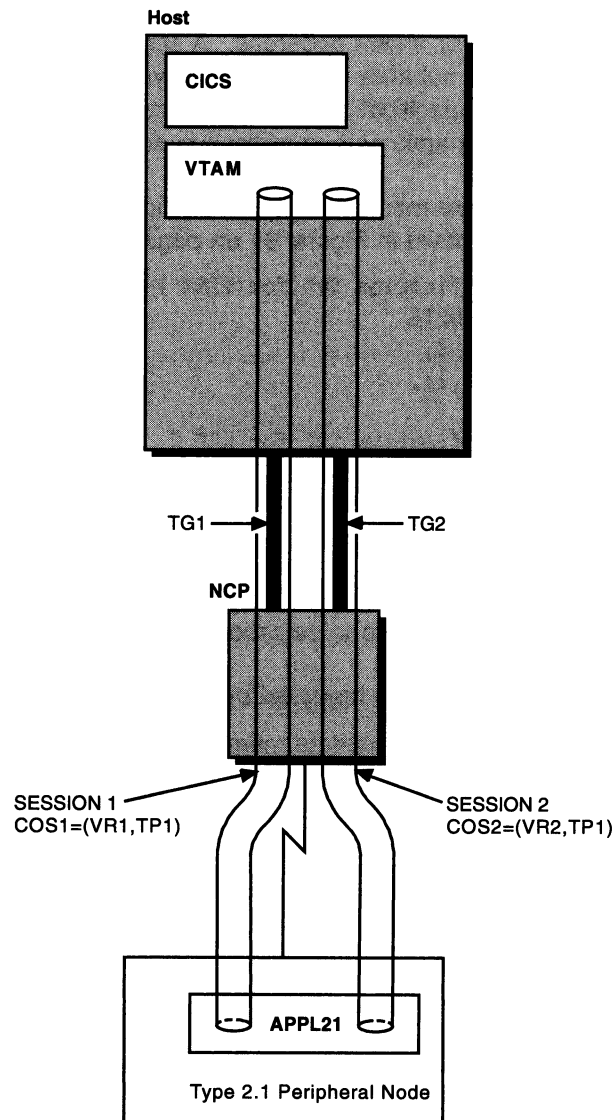


Figure 80. Parallel Sessions Using Parallel Transmission Groups

Dynamic Path Update

Dynamic path update enables the VTAM operator to add NCP path definitions dynamically without regenerating and reloading the NCP. Using the same operator facilities, you can also dynamically add VTAM path definitions. The explicit route definitions in VTAM and NCP can be replaced or deleted only if the explicit route is inoperative (a status of INOP).

Note: If you are not attempting to modify an ER's mapping to a TG, the ER state is not checked and the ER can be either operative or inoperative.

Dynamic path update can also be used to change path definition values for virtual route pacing window size on inoperative routes and transmission group thresholds on operative routes.

To use dynamic path update, at least one route must be defined and operative to establish the necessary SSCP-PU session between VTAM and each NCP. The dynamic path update changes are transmitted to NCP on this session. Dynamic path update members are filed in the VTAM definition library. A dynamic path

update member can contain path specifications that are added, replaced, or deleted from one or more NCP or VTAM subareas. A set of path update specifications that is targeted to a specific subarea node within a dynamic path update member is called a dynamic path specification set. A dynamic path update member can contain one or more dynamic path specification sets.

The dynamic path update member in the following sample can be used to add NCP3 to the network shown in Figure 81 on page 315.

```
* Dynamic Path Specification Set for VTAM 1 (subarea 1)
VTAM1  VPATH NETID=NETA
        PATH  DESTSA=3,
            ER0=(2,1),
            VR0=0
* Dynamic Path Specification Set for NCP 2 (subarea 2)
NCP2    NCPPATH NETID=NETA
        PATH  DESTSA=3,
            ER0=(3,10)
```

Note that the name on the NCPPATH definition statement identifies the NCP for which the PATH statements are to be processed.

A dynamic path update can be triggered by any of the following methods:

- Activating a dynamic path update member using the VTAM configuration list or the VARY operator command. An active SSCP-PU session must exist between VTAM and the NCP whose routes are being dynamically updated, and the explicit routes being changed must be inoperative.
- Specifying the NEWPATH operand when activating an NCP with an operator command. This causes VTAM to activate the dynamic path update member immediately following NCP activation.
- Activating an NCP that has the NEWPATH operand specified on the PCCU definition statement for this host. This causes VTAM to activate the dynamic path update member immediately following NCP activation.

Note: Because dynamic path update changes do not alter VTAM or the NCP load module, you should use a method that ensures that these changes are reinstated if VTAM is restarted or if NCP is reloaded.

The dynamic path update member in the preceding example could be filed in VTAM1's definition library. When the operator activates the dynamic path update member, VTAM1 identifies the dynamic path update member because the name in the VPATH definition statement matches its SSCPNAME (VTAM1). VTAM1 transmits the dynamic path update member to NCP2 using the active session path (SSCP-PU session).

Note: You should file the dynamic path update member for each subarea in separate dynamic path update members to avoid possible performance deterioration.

The following restrictions are placed on coding dynamic path update members and dynamic path specification sets:

- A dynamic path update member can include path definitions for only one VTAM host. If an invalid definition statement or VPATH or NCPPATH definition statement is encountered, VTAM discards all PATH definition statements that follow the invalid definition statement up to the next recognizable VPATH or

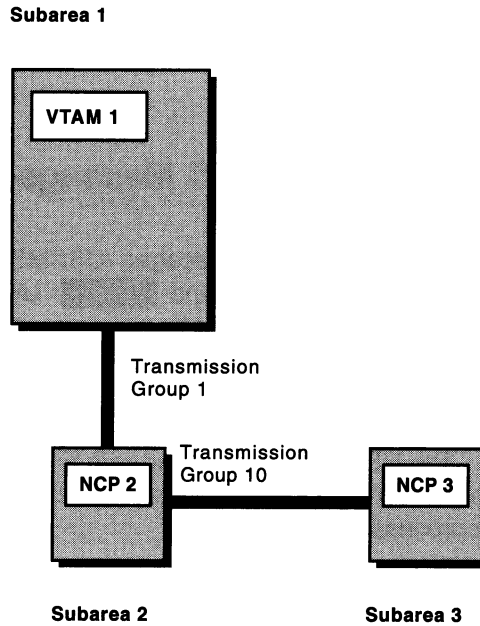


Figure 81. Sample Single-Domain Network

NCPPATH definition statement and issues a warning message. PATH definition statements that were already processed are not affected.

- When a dynamic path update member is activated using the VTAM configuration list or the VARY operator command, an NCPPATH definition statement is considered to be invalid if the NCP name on the NCPPATH definition statement does not match the resource name of an active NCP.
- When a dynamic path update member is activated by specifying the NEWPATH operand, an NCPPATH definition statement is considered to be invalid if the NCP name on the NCPPATH definition statement does not match the resource name of the NCP being activated.
- A dynamic path update member can include the DELETER operand if it applies to the host activating the dynamic path update member.

MVS, VSE VTAM saves the name of the dynamic path update member in the NODELST data set (VTAM's configuration restart data set). The actual dynamic path specification sets are not saved in a VSAM configuration restart file.

Note: Extreme caution must be used when deleting or replacing a route for a node that is an intermediate routing node for other routes.

The NCPPATH definition statement identifies the NCP and the network to which the dynamic path update member applies in a dynamic path update member.

Establishing and Controlling SNA Sessions

This section contains the following subsections:

- National Language Support for End User USS Messages and Commands
- National Language Support for the Language Passed to Application Programs
- Model Name Table
- Associated LU Table
- Automatic Logons

- Session Authorization and Accounting
- Pacing and Flow Control
- Application Program Pacing
- Setting Pacing Values.

National Language Support for End-User USS Messages and Commands

Without national language support, USS messages (USSMSG00–USSMSG14) and commands (LOGON, LOGOFF, IBMTEST, and **VM,VSE** UNDIAL) came from either a USS table that was statically defined to a logical unit, or the IBM-supplied default table, ISTINCNT. With national language support, end users can choose a USS table by specifying the LANGTAB operand on any valid USS command. This enables end users to enter USS commands and receive USS messages in the language of their choice. The USS table associated with the logical unit using the LANGTAB operand remains in effect until another valid USS command with the LANGTAB operand is processed.

MVS You can also define USS messages using the MVS message service (MMS), which enables you to define each USS message in several different languages. End users can specify which language to use to retrieve USS messages from the MMS by using the LANG parameter on any valid USS command. The language for USS messages associated to the logical unit using the LANG parameter remains in effect until another valid USS command with the LANG parameter is processed.

For example, USSMSG10 can be defined in multiple languages, using the LANGTAB USS tables or MMS.

Remember that the language specified for the end user USS commands (from the LANGTAB operand) and messages (from the LANG or LANGTAB operands) remain in effect for a logical unit (terminal) after the end user logs off an application program. One logical unit (terminal) might be used by several users who speak different languages. Because LANG and LANGTAB can be entered with null values to cancel their effects, it might be wise to define LOGON commands with default null values for LANG and LANGTAB. For example, you can code the following as a default for the LOGON command:

```
LOGON    USSCMD    CMD=LOGON,FORMAT=PL1
          USSPARM   PARM=APPLID
          USSPARM   PARM=LANG,DEFAULT='()'
          USSPARM   PARM=LOGMODE
          USSPARM   PARM=DATA
          EJECT
```

Note: The coding for LANGTAB is the same as shown in the example for LANG.

Defining USS Tables for Use with the LANGTAB Operand (MVS)

The following restrictions apply when defining a USS table used for LANGTAB:

- The name of the USS tables you code must match the name the end user specifies on the LANGTAB operand.
- The name of the USS table cannot be ISTINCNO or ISTCFCMM.
- FORMAT=DYNAMIC must be coded on the USSTAB macroinstruction to use the table for LANGTAB.

- If TABLE is coded on the USSTAB macroinstruction, it is ignored because the input translation table cannot be specified by the end user.
- USS message text must comply with the rules detailed in “Unformatted System Services (USS) Tables” in the *VTAM Resource Definition Reference*.

Defining USS Messages to the MVS Message Service

To define USS messages to MMS:

1. Create message skeleton files for MMS. These files define the end user messages (USSMSG00 – USSMSG14) in the languages required for your installation.
2. Define data sets to MVS.
3. Run the message compiler.
4. Update SYS1.PARMLIB.
5. Activate MMS.

For more information on message translation using MMS, see Appendix G, “Message Translation Using the MVS Message Service” on page 681. For information on performing these functions, see *MVS/ESA Planning: Operations*. For information on the format of message skeleton files used for language translation, see *MVS/ESA Application Development Guide: Assembler Language Programs*.

When defining a USS message in a message skeleton file, use the substitute tokens &1., &2., and &3. for the first, second, and third variable data fields for the message. These tokens correspond to the %(1), %(2), and %(3) syntax of specifying variable data that is used for the TEXT operand of the USSMSG macroinstruction.

National Language Support for the Language Passed to Application Programs (MVS)

By default, the language passed to a VTAM application program comes from the logon mode entry specified by the USS LOGMODE parameter. With national language support, the language passed to the application program can be specified by the end user, independent of the LOGMODE parameter.

The end user can enter a LOGON command to indicate which language is to be passed to the application program using:

- The language code parameter of the LANGTAB operand
- The LANG operand.

Notes:

1. The language that is passed to the application program does not remain in effect across logons (as it does for USS messages). Each LOGON command must specify one of the above (LANG operand or the language code parameter of the LANGTAB operand) to have a language passed to the application; otherwise, the default (MODEENT LANG operand value) is used.
2. The LANG operand is valid only for MVS. For VM and VSE systems, the LANG operand is accepted, but is ignored.
3. If both operands indicate a language, the LANG operand has priority.

4. If the MMS is not currently active, the LANG operand is ignored.

For more information about the LANG and LANGTAB operands, see “LANG (MVS)” and “LANGTAB” in the *VTAM Resource Definition Reference*. For specific information on how the application program receives the specified language, see “Control Blocks Used for Session Establishment and Termination” in *VTAM Programming*.

Model Name Table

You create a model name table by specifying the following:

- A MDLTAB macroinstruction
- One or more MDLENT macroinstructions
- Optional MDLPLU macroinstructions for each MDLENT.

File these macroinstructions in the VTAM definition library. The name of the table is the name of the stored member or file. No assembly or link-edit is required to install the table. The table is built dynamically upon the first activation of any resource that has a defined association to the table.

IBM does not supply a default model name table.

An example of a model name table is shown in Table 21.

Table 21. Example of Model Name Table

Name	Operation	Operands
MDLTAB1	MDLTAB	
ENTRY1	MDLENT	MODEL=MDLNAME1
ENTRY2	MDLENT	
	MDLPLU	PLU=APPL2, MODEL=MDLNAME2
	MDLPLU	PLU=APPL5, MODEL=MDLNAME3
ENTRY3	MDLENT	MODEL=MDLNAME3
	MDLPLU	PLU=APPL5, MODEL=MDLNAME4
	MDLPLU	PLU=APPL7

Following are some examples of how the model name table in Table 21 works:

- Assume that the SLU has the MDLTAB=MDLTAB1 macroinstruction coded in its definition. The entry name specified by the SLU and the name of the PLU involved in the session determines which model name is sent to the PLU during session initiation.
- Assume the SLU has the MDLENT=ENTRY1 macroinstruction coded in its definition. This macroinstruction specifies a default model name of MDLNAME1 for the entry. Because no MDLPLU macroinstructions are specified, model name MDLNAME1 is sent for any PLU.
- Assume the SLU has the MDLENT=ENTRY2 macroinstruction coded in its definition. This macroinstruction specifies no default model name. The first MDLPLU macroinstruction specifies that model name MDLNAME2 is sent when the PLU is APPL2. The second MDLPLU macroinstruction specifies that model name MDLNAME3 is sent when the PLU is APPL5. For any other PLU, no model name is sent.
- Assume the SLU has the MDLENT=ENTRY3 macroinstruction coded in its definition. This macroinstruction specifies a default model name of MDLNAME3 for

the entry. The first MDLPLU macroinstruction specifies that model name MDLNAME4 is sent when the PLU is APPL5. The second MDLPLU macroinstruction specifies that no model name is sent when the PLU is APPL7. For any other PLU, model name MDLNAME3 is sent.

Associated LU Table

You create an associated LU table by specifying the following:

- An ASLTAB macroinstruction
- One or more ASLENT macroinstructions
- Optional ASLPLU macroinstructions for each ASLENT.

File these macroinstructions in the VTAM definition library. The name of the table is the name of the stored member or file. No assembly or link-edit is required to install the table. The table is dynamically built upon the first activation of any resource that has a defined association to the table.

IBM does not supply a default associated LU table.

Note: A printer must be in the same network as the terminal or must have a unique name that requires no translation across network boundaries. Violation of this restriction can cause failure of print-screen operations during cross-network sessions.

An example of an associated LU table is shown in Table 22.

Table 22. Example of Associated LU Table

Name	Operation	Operands
ASLTAB1	ASLTAB	
ENTRY1	ASLENT	PRINTER1=PTRA,PRINTER2=PTRB
ENTRY2	ASLENT	
	ASLPLU	PLU=APPL2, PRINTER1=PTRC,PRINTER2=PTRD
	ASLPLU	PLU=APPL5,PRINTER1=PTRA
ENTRY3	ASLENT	PRINTER1=PTRA
	ASLPLU	PLU=APPL2,PRINTER1=PTRC,PRINTER2=PTRD
	ASLPLU	PLU=APPL7,PRINTER2=PTRB

Assume that the SLU has the ASLTAB=ASLTAB1 macroinstruction coded in its definition. The entry name specified by the SLU and the name of the PLU involved in the session determines which printer names are sent to the PLU during session initiation. For example:

- Assume the SLU has the ASLENT=ENTRY1 macroinstruction coded in its definition. This macroinstruction specifies a default primary printer name of PTR A and a default alternate printer name of PTRB for the entry. Because no ASLPLU macroinstructions are specified, primary printer name PTR A and alternate printer name PTRB are sent for any PLU.
- Assume the SLU has the ASLENT=ENTRY2 macroinstruction coded in its definition. This macroinstruction specifies no default printer names for the entry. The first ASLPLU macroinstruction specifies that primary printer name PTRC and alternate printer name PTRD are sent when the PLU is APPL2. The second ASLPLU macroinstruction specifies that primary printer name PTR A is sent and no alternate printer name is sent when the PLU is APPL5. For any other PLU, no printer names are sent.

- Assume the SLU has the ASLENT=ENTRY3 macroinstruction coded in its definition. This macroinstruction specifies a default primary printer name of PTR A for the entry. The first ASLPLU macroinstruction specifies that primary printer name PTR C and alternate printer name PTR D are sent when the PLU is APPL2. The second ASLPLU macroinstruction specifies that no primary printer name is sent and alternate printer name PTR B is sent when the PLU is APPL7. For any other PLU, primary printer name PTR A is sent and no alternate printer name is sent.

Selecting Session Parameters for the Logon Mode Table

To create the logon mode table:

- Modify the IBM-supplied table.
- Create supplementary tables and associate them with device-type LUs or application programs using the MODETAB operands in the definition statements defining them. You can use the DLOGMOD operand on the appropriate definition statements to indicate the specific logon mode entry in the table that contains the session parameters to be used. The LOGMODE operand can also be specified on VARY LOGON commands to specify a logon mode table entry.

Figure 82 shows the sequence in which MODETAB, MODEENT, and MODEEND macroinstructions are coded to define a logon mode table.

Figure 83 on page 324 shows a summary of the algorithm VTAM uses to obtain session parameters.

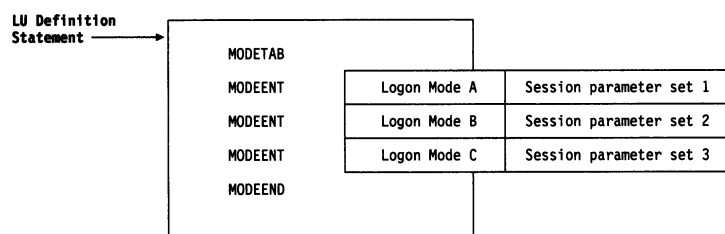


Figure 82. Macroinstructions for Logon Mode Table

Logon mode tables are created or modified using MODETAB, MODEENT, and MODEEND macroinstructions. “Logon Mode Table” in the *VTAM Resource Definition Reference* describes the contents of the IBM-supplied logon mode table.

MVS It also describes how TSO can query for the logon mode being used by a terminal in place of supplying the LOGMODE during the logon.

The various ways of specifying session parameters are illustrated in Table 23 on page 321.

Table 23 (Page 1 of 2). How Session Parameters Are Identified

Request Source	Request Form	How Session Parameters Are Identified	
		Logon Mode Table Entry	Logon Mode Table
Dependent Logical Unit	- Initiate Self request - Interpret table character string representing logon - Character-coded logon	- Field in Initiate Self request containing mode name used to search logon mode table - Parameter in a character-coded logon that translates into LOGMODE (name) - DLOGMOD operand on LU definition statement - Default entry ISTCOSDF, if the ISTCOSDF start option allows its use for dependent LUs - First entry in logon mode table named in MODETAB operand on LU definition statement - First entry in IBM-supplied logon mode table	- MODETAB operand on LU definition statement - IBM-supplied logon mode table.
Independent Logical Unit	- BIND - Cross-domain initiate request	- Field in BIND or cross-domain initiate request containing mode name used to search logon mode table - DLOGMOD operand on CDRSC definition statement - Default entry ISTCOSDF, if the ISTCOSDF start option allows its use for independent LUs - First entry in logon mode table named in MODETAB operand on CDRSC definition statement - First entry in IBM-supplied logon mode table	- MODETAB operand on CDRSC definition statement - IBM-supplied logon mode table.

Table 23 (Page 2 of 2). How Session Parameters Are Identified

Request Source	Request Form	How Session Parameters Are Identified	
		Logon Mode Table Entry	Logon Mode Table
Application Program	Application program issuing VTAM macroinstructions such as SIMLOGON or REQSESS	- Name of logon mode table entry specified in the NIB - DLOGMOD operand on LU definition statement - Default entry ISTCOSDF, if the ISTCOSDF start option allows its use for application programs - First entry in logon mode table named in MODETAB operand on LU definition statement - First entry in IBM-supplied logon mode table	
Automatic logon	Initiate Other request	- DLOGMOD operand on LU definition statement - Default entry ISTCOSDF, if the ISTCOSDF start option allows its use - First entry in logon mode table named in MODETAB operand on LU definition statement - First entry in IBM-supplied logon mode table	
VTAM operator	VARY LOGON command	- LOGMODE operand in VARY LOGON command - LOGMODE operand in previous VARY LOGON command - DLOGMOD operand on LU definition statement - Default entry ISTCOSDF, if the ISTCOSDF start option allows its use - First entry in logon mode table named in MODETAB operand on LU definition statement - First entry in IBM-supplied logon mode table	

Regardless of the source of the logon or what session parameters are associated with the logon, the application program decides which session parameters are used for the session. It can decide to use the session parameters associated with the pending logon, or it can choose a different set of parameters. When the application program issues a macro to initiate a session with a logical unit, the application

program indicates the logon mode name or session parameters to be used during the session. The logon mode table from which the session parameters are selected is always the one associated with the SLU, or, if a table is not associated with the logical unit, it is the IBM-supplied default logon mode table in the VTAM that owns the SLU.

If the specified logon mode entry does not exist in the logon mode table associated with the secondary logical unit, the default logon mode (ISTCOSDF) is used if it is found in the table and its use is allowed as determined by the ISTCOSDF start option.

You can modify or replace the IBM-supplied logon mode table, provided that the modified or replacement table has the same name as the IBM-supplied table and that the IBM-supplied table is deleted. However, because the IBM-supplied table might be needed for problem determination, you should create supplementary tables instead of deleting the IBM-supplied table.

As part of the session-establishment procedures, an application program acting as the PLU can:

- Supply a logon mode name associated with the SLU. The session parameters associated with the logon mode name in the domain of the SLU are used.
- Directly supply the session parameters to be used.
- Use the session parameters associated with the pending logon.

Logon mode entries from MVS or VSE do not provide the necessary logon information to a VM host when connecting to VM unless the device type is a 3277.

“How Logon Mode Names and Session Parameters Are Used” in *VTAM Programming* describes in more detail how the application program handles session parameters.

Automatic Logons

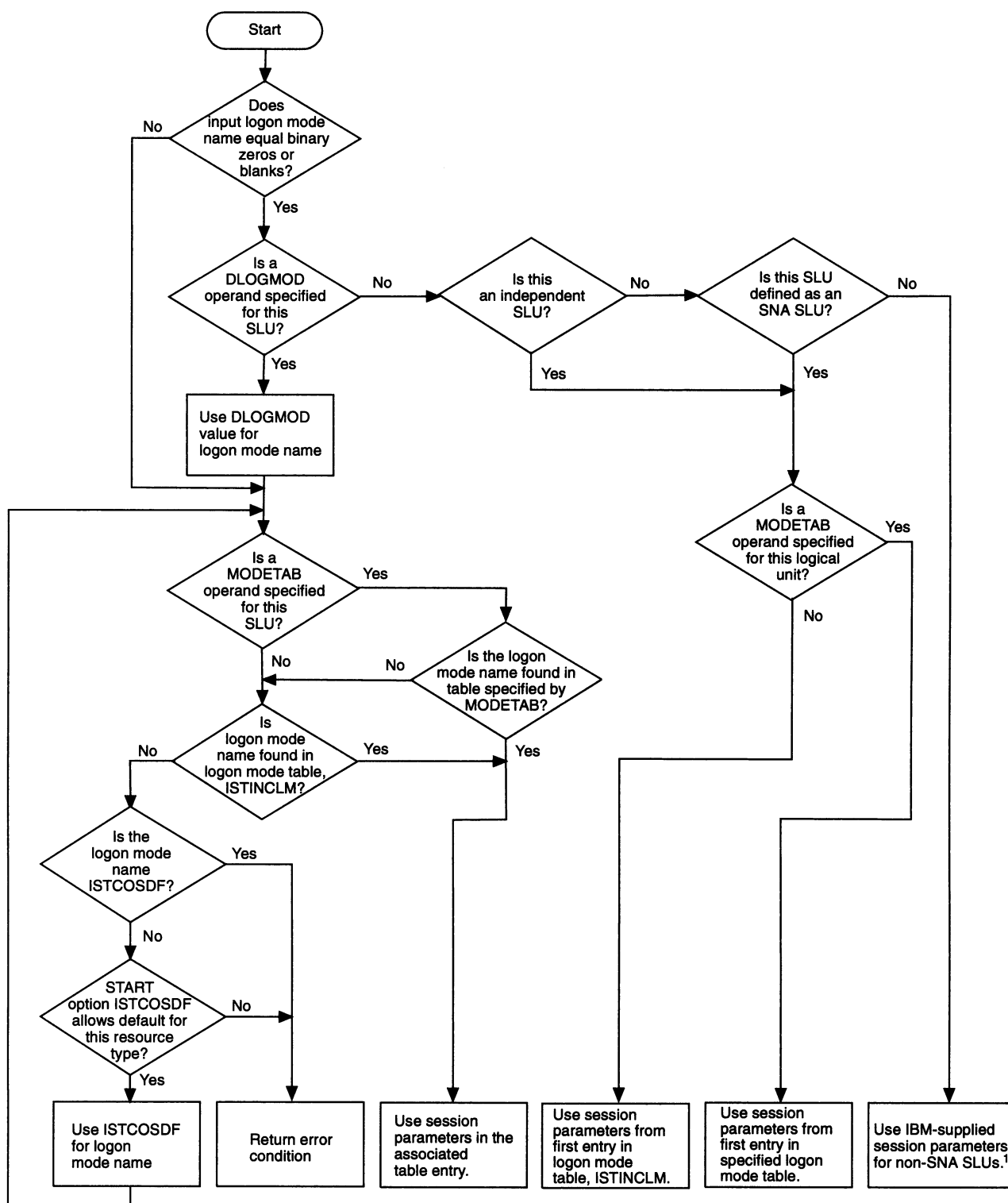
This section contains the following subsections:

- Coding for Automatic Logon
- Operator Commands for Automatic Logon
- Reallocation of Autologon Sessions.

Coding for Automatic Logon

By using the LOGAPPL operand on the definition statement for an LU, the LU can automatically initiate a session with a specific application program whenever the LU is activated. The application program can either be the program with which the LU ultimately initiates a session or an intermediary application program to which the LU is temporarily assigned. For application programs in another network, the network-qualified name can be specified on the LOGAPPL operand.

The application program to which an LU is automatically logged on is called its controlling application program. If the LU is passed to another application program, VTAM automatically initiates a session with the controlling application program when that session is terminated. If the application program with which the LU is in session terminates that session and there is no controlling application program, the LU is available for session initiation.



¹ VTAM uses the following session parameters for the non-SNA SLU:

MODEENT LOGMODE=NONSNA, FMPPROF=X'02', TSPROF=X'02', PRIPROT=X'71', SECPROT=X'40', COMPROT=X'2000'

Figure 83. How Session Parameters Are Obtained from a Logon Mode Table

Figure 84 on page 325 shows a sample automatic logon session:

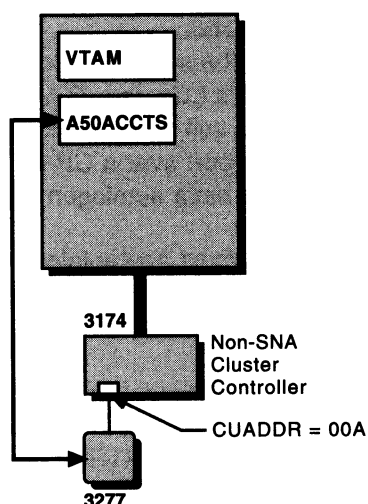


Figure 84. Automatic Logon to A50ACCTS Application Program

Following is the sample coding for the A50ACCTS application program in Figure 84:

```
APPLNODE VBUILD TYPE=APPL
A50ACCTS APPL  ATNLOSS=ALL,      ENABLE ATTN EXIT FOR ALL LU 6.2 SESSIONS
                APPC=YES,        ENABLE LU 6.2 SUPPORT
                PARSESS=YES,     ENABLE PARALLEL SESSION FOR LU 6.2 SUPPORT
                SECACPT=ALREADYV, ACCEPT ALREADY VERIFIED LU 6.2 CONVERSATIONS
                ACBNAME=ACCOUNTS NETWORK LU NAME=A50ACCTS
                                VTAM APPLICATION APPLID=ACCOUNTS

:
```

Following is the sample coding for the 3277 display station in Figure 84:

```
LC3270NS LBUILD          LOCAL NON-SNA ATTACHMENT
LOC3277  LOCAL  CUADDR=00A, CHANNEL UNIT ADDRESS
                TERM=3277,   TERMINAL TYPE

:

                DLOGMOD=S3270,  DEFAULT LOGON MODE TABLE ENTRY
                LOGAPPL=A50ACCTS AUTOMATIC LOGON APPLICATION
```

Operator Commands for Automatic Logon

You can establish an automatic logon session using the VARY LOGON command.

After the automatic logon is established, it remains until one of the following occurs:

- The major node of one of the session partners is deactivated.
- The primary logical unit is changed using the VARY LOGON or VARY ACT,LOGON command. If the primary logical unit is a USERVAR name, the session setup requests can be rerouted to a different primary logical unit by using the MODIFY USERVAR command.
- The automatic logon is deleted using the VARY NOLOGON command.

You can view the automatic logon relationship by using the DISPLAY ID command.

Reallocation of Autologon Sessions

For dependent LUs, the VTAM that owns the secondary logical unit controls the automatic logon sessions. For independent LUs, the VTAM providing boundary function services for the secondary logical unit controls the automatic logon sessions. VTAM also provides autologon support over a CP-CP session for secondary LUs whose network node server does not have autologon capabilities.

By default, adjacent CDRM activation re-drives any autologon sessions that are not established between resources of the two VTAMs. Adjacent CP activation also re-drives any such autologon sessions, if the adjacent CP supports automatic logon.

To specify which, if any, adjacent node activations are to result in retrying pending autologon requests, use the AUTORTRY start option. In addition to the default for AUTORTRY described above, you can use AUTORTRY to limit session retries to only after adjacent CDRM activation (AUTORTRY=CDRM), to suppress all retries for adjacent node activations (AUTORTRY=NONE), or to specify that all adjacent node activations are to result in session retry (AUTORTRY=ALL). For AUTORTRY values other than NONE, CDRM deactivation causes re-drives of any autologon sessions that relied on the lost CDRM. CP deactivation does not cause re-drives.

Use the AUTOTI start option to control how often pending autologon requests are retried. Time values are specified in seconds, minutes, hours, or days. If AUTOTI is non-zero, re-drives occur at timer expiration for all autologon sessions that are not established, regardless of the value of AUTORTRY. Use AUTOTI to ensure session setup in the following situations:

- Multi-hop CP-CP network
- Network node server has not yet received the controlling primary LU's registration
- CLSDST PASS to an unavailable application
- USERVAR value changed while a node is inactive, and the node is then re-activated.

Note: Using SRTIMER, SRCOUNT, or both in conjunction with AUTOTI can produce unexpected results. For information on the interaction of AUTOTI with SRTIMER and SRCOUNT, see “Improving VTAM Performance Using Start Options” on page 131.

Sample Automatic Logon Reallocation: In Figure 85 on page 327 assume that secondary logical unit LU1A is in session with an application program APPL1, and that LU1A has LOGAPPL=A50ACCTS coded. When the session with APPL1 ends, VTAM attempts to establish a session between LU1A and A50ACCTS.

If A50ACCTS is inactive, the automatic logon session fails. If A50ACCTS's major node is active to VTAM but it has not opened its ACB yet, the automatic logon session fails, but is initiated when A50ACCTS opens its ACB.

If you use the VARY INACT command to deactivate A50ACCTS, you can only regain the automatic logon sessions with LU1A and LU2A by issuing a VARY ACT,LOGON command for A50ACCTS or reactivating A50ACCTS and then reactivating LU1A and LU2A.

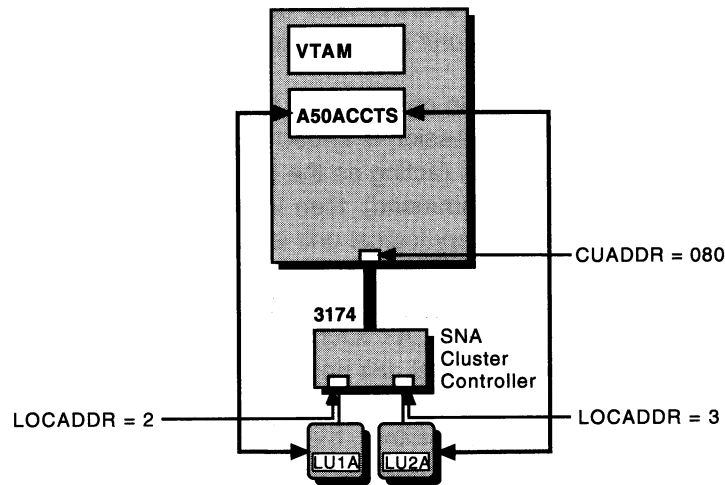


Figure 85. Automatic Logon Reallocation

Conditions for Reallocation: A session is reallocated as a result of one of the following conditions:

- If a secondary logical unit goes through a state change from disabled or inhibited to enabled, then the owning VTAM attempts to set up a session with the controlling application program for the secondary logical unit. The session state is determined by the device and is indicated to VTAM by an ACTLU response from the logical unit or a NOTIFY request from the device.
- If a session is terminated, VTAM checks to see whether an automatic logon relationship exists and, if so, reallocates that session if the session being ended is not with the controlling application program.
- If the controlling application program receives a termination request with TYPE=FORCE to terminate the controlling session, the controlling session will be reestablished if a session path exists at that time between the two LUs.

Note: If a termination request is issued with TYPE=UNCOND and flows through an APPN network towards the PLU, the request is promoted to TYPE=FORCE. To avoid re-driving the session, issue VARY TERM, TYPE=COND. For further information, see "VARY TERM Command" in *VTAM Operation*.

How Primary Logical Unit Status Affects Reallocation: VTAM attempts to start a session with the controlling application program (the primary logical unit). Following is an explanation of how VTAM reacts to the primary logical unit's status:

- If the primary logical unit cannot be found (the major node is not active) or is inactive, the automatic logon session fails and is not automatically reallocated when the primary logical unit is activated.
- If the primary logical unit is in CONCT status (active to VTAM but OPEN ACB has not been issued), the automatic logon fails with sense code 088A0003 and VTAM tries to set up the session again when the primary logical unit issues an OPEN ACB and SETLOGON OPTCD=START macros.
- If the primary logical unit is deactivated through VTAM by an operator command and becomes inactive, the secondary logical unit must re-attempt the automatic logon session using a secondary logical unit state change, or the

operator can issue a VARY ACT command for the secondary logical unit after the primary logical unit is at least in CONCT status.

The AUTOTI start option can be used to cause re-drives to attempt session setup. If an automatic logon session is ended normally (with a CLSDST command) and the application program (acting as the primary logical unit) is taken down normally (with a CLOSE ACB command), then VTAM reestablishes the automatic logon session when the primary logical unit is restarted.

Determining Automatic Logon Relationships: To determine an automatic logon relationship, use the DISPLAY ID=(*SLUname*),E command. Use this command to determine whether a logical unit has a controlling primary logical unit. Message IST1131I contains the name of the primary logical unit. Message IST635I contains the name of the session partner, the status of the session, and other session information. The status of the session will have the last two characters overlaid with /C if this session is due to an automatic logon relationship.

Sample Displays Received during Reallocation: If the primary logical unit is in CONCT status when the automatic logon session set up fails, you will receive the following messages:

```
IST663I INIT OTHER OR CDINIT REQUEST FAILED, SENSE=088A0003
IST664I REAL OLU= <SLU-NAME> REAL DLU= <PLU-NAME>
IST889I SID= <SESSION ID>
IST890I AUTOLOGON SESSION SETUP FAILED
IST896I AUTOLOGON WILL BE RETRIED WHEN CONTROLLING PLU IS AVAILABLE
IST314I END
```

Note: Besides normal message suppression, the number of IST663I and IST664I messages for the 088A0003 sense codes can be suppressed using the PLUALMSG and SLUALMSG start options. For more information, see Chapter 4, "Start Options" in the *VTAM Resource Definition Reference*.

If the primary logical unit was in CONCT status when the automatic logon failed and then issues an OPEN ACB, you will receive the following messages:

```
IST899I RETRY OF AUTOLOGON(S) TO <PLU-NAME> IN PROGRESS
```

If the primary logical unit is not active when the automatic logon fails, you will receive the following messages:

```
IST663I INIT OTHER OR CDINIT REQUEST FAILED, SENSE=08570002
IST664I REAL OLU= <SLU-NAME> REAL DLU= <PLU-NAME>
IST889I SID=.....
IST890I AUTOLOGON SESSION SETUP FAILED
IST1138I REQUIRED RESOURCE <PLU-NAME> NOT ACTIVE
IST314I END
```

Note: The sense code in message IST663I and the text of message IST1138I might be different.

If you had automatic logon sessions fail on a primary logical unit that was in CONCT status and then you deactivate it before it opens its ACB, you will receive the following messages:

```
IST097I VARY ACCEPTED
IST899I RETRY OF AUTOLOGON(S) TO <PLU-NAME> WILL NOT OCCUR
IST1133I <PLU-NAME> NODE NOW INACTIVE
```

Session Authorization and Accounting

VTAM calls the session management exit routine whenever a session between logical units is being established, ended, taken over, or recovered. The session management exit routine is invoked by every VTAM in the session-setup path. It is also called for session accounting when an LU-LU session ends.

Note: Improper use of the session management exit routine can impact VTAM operation. For example, you should consider the following:

- A program error can affect VTAM, the operating system, and other programs. Although abnormal end recovery is provided for errors within the exit, the exit code can accidentally corrupt VTAM storage.
- Lengthy processing time could degrade VTAM performance. While these routines are running, VTAM cannot process any new VTAM operator requests to initiate or end sessions. To reduce the effect on VTAM's performance, any analysis of the information collected should be done later by another program.

VTAM supports session authorization exit routines, which can be used to control session initiation. These routines can accept or reject the session based on various information that is passed to them. VTAM also supports session accounting exit routines that can be used to perform accounting functions for established sessions.

The session authorization and session accounting exit routines provide some of the same functions as the session management exit routine. These exits are invoked prior to the corresponding session management function. However, these exit routines do not provide all of the functions that are available in the session management exit routine, and they are not recommended for VTAM V2R2 or later releases.

For further information on the session management, session authorization, and session accounting exit routines, including a detailed description of the vector lists and the information provided to the routines, see "Session Management Exit Routine," "Session Authorization Exit Routine," and "Session Accounting Exit Routine" in *VTAM Customization*.

Session Authorization

When SSCP routing is in process for cross-domain sessions, full information about the destination logical unit might not be available when the session management exit routine is first invoked. To enable proper authorization to be done for cross-domain sessions, two separate calls for authorization can be made for these sessions. The first call is made early in the session setup process. The session management exit routine can accept or reject the session at this time, or it can defer its decision until later. If the decision is deferred, the routine is called again for a secondary authorization after full information is known about the destination logical unit.

You might code the routine to contain a table of valid sessions against which the session-establishment request can be compared. For example, an application program can be designed to establish a session with any logical unit using the OPNDST macroinstruction with the ACCEPT option in its LOGON exit routine. The authorization exit routine can compare the identity of any logical unit that attempts to establish a session with the application program to entries in such a table to determine whether authorization can be granted for that logical unit.

Both primary and secondary authorization exits are also driven during LU takeover processing if the sessions being taken over were established using extended BIND protocols.

Session Accounting

With the capability of peripheral LUs serving as primary LUs (PLUs), accounting application programs relying on the primary LU to be located only in the host might not be sufficient.

At LU-LU session setup time, a session awareness PIU is sent to the NetView Performance Monitor (NPM) as unsolicited information. Byte and PIU counts are updated every time there is traffic on the session. When the counts surpass the user-specified threshold, the counters are sent to NPM. At the end of the session, session awareness data and the last set of counters for the session are sent to NPM. The user-specified options as well as thresholds can be changed using NPM commands. In addition, transport of accounting data to NPM can be suspended and resumed using NPM commands.

If selecting session accounting, the user can specify the following during NCP generation:

- Whether session accounting is required for PLUs, SLUs, or all LUs
- Whether accounting collection should begin immediately or be deferred
- The session accounting byte and PIU thresholds
- Whether backup NPM sessions are defined
- Whether gateway session accounting is to be used for cross-network sessions
- The number of half-sessions for which session accounting will be done
- The number of session accounting extension blocks to define
- The PIU distribution ranges (if any) to be used.

Pacing and Flow Control

This section contains the following subsections:

- Session Pacing
- Fixed Session Pacing
- Overriding Fixed Pacing Counts
- Adaptive Session Pacing.

Session Pacing

Session pacing controls the number of normal-flow requests allowed to flow at one time between the sender and the receiver in a session. The receiver can control the rate at which the sender sends requests and thus can better control the use of its own resources. For example, pacing helps the receiver control the allocation of its buffer resources. Pacing applies independently to requests flowing from the primary to the secondary and from the secondary to the primary ends of a session.

There are two types of session pacing: fixed and adaptive. With fixed session pacing, the amount of data that can be transmitted or received between session partners is maintained as a constant window size. With adaptive session pacing, the window size can vary depending on the resources available at each session endpoint; that is, after each window of data is transmitted, the receiver informs the

transmitter how large the next window can be. For peripheral nodes, you can code the PACING and VPACING operands to support either fixed or adaptive session pacing for sessions with that node. Whether adaptive session pacing is supported is determined by the capability of the physical units. For more information on coding these values, see the *VTAM Resource Definition Reference*.

Pacing between session endpoints occurs in stages depending on the type of session pacing used (fixed or adaptive), the direction of the transmission (primary to secondary or secondary to primary), and the subarea nodes that are performing boundary function.

In one-stage pacing, the pacing occurs directly between the two LUs involved in the session. For example, one-stage pacing is used when two VTAM application programs are in session. In two-stage pacing, the pacing occurs between one of the LUs and a boundary function and then between the boundary function and the other LU in the session. In Figure 86, APPL1 is in session with a PU type 2. Two-stage pacing occurs from APPL1 (primary logical unit) to the PU type 2 (secondary logical unit), and one-stage pacing occurs from the PU type 2 to APPL1. In this case, the NCP in the communication controller provides the boundary function for the LU.

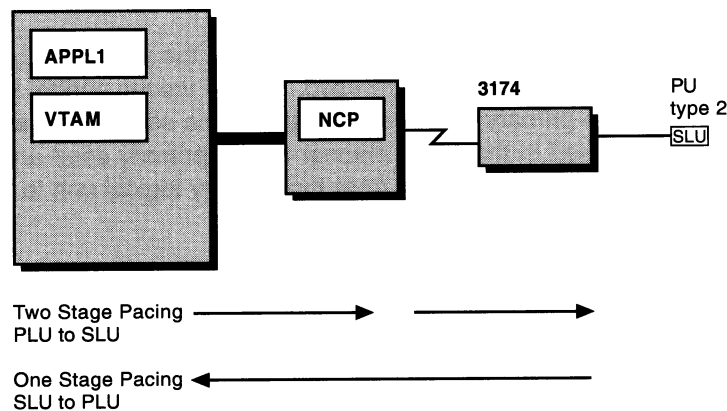


Figure 86. One- and Two-Stage Pacing

The number of pacing stages can be the same for both directions (primary-secondary and secondary-primary) in a session, or it can be different. In the previously mentioned example of two application programs being in session, one-stage pacing is used for both directions. You must have NCP Version 5 Release 2 or higher to support two stage and adaptive session pacing involving a 3745 communication controller or Version 4 Release 3 or higher for a 3725 communication controller.

When pacing is in effect, the sender for a given pacing stage in a session is allowed to send a limited number of normal-flow requests. The sender is not allowed to send any more such requests until the receiver indicates that it is ready to receive more by returning a pacing response to the sender. The sender can then send more requests until it must again stop and wait for another pacing response. If the receiver sends the pacing response early (that is, before having received the maximum number of requests), the sender completes sending the first group of requests and can then send another group of requests. The pacing of requests flowing toward a given LU (including an application program) is called inbound pacing for that LU. Similarly, the pacing of requests flowing away from a logical unit is called outbound pacing for that LU.

Fixed Session Pacing

A separate pacing window value can be specified for each of the four possible pacing stages in a session. The stages and pacing windows used in each stage are illustrated in Figure 87 on page 333.

The values are carried in the SNA Bind Session request from the primary LU to the boundary function (if one exists for this session), then to the secondary LU. In this way, all the participants in the session are informed of the pacing values to be used. The possible pacing windows for a session are the following:

- Primary send (PS)
- Secondary receive (SR)
- Secondary send (SS)
- Primary receive (PR).

When data is sent from the PLU to the SLU, the PS and SR values are used. The primary send count controls the flow of requests from the primary logical unit to the boundary function, and the secondary receive count controls the flow from the boundary function to the secondary logical unit. For one-stage pacing, the primary send and secondary receive counts are the same. For two-stage pacing, they can be different.

When data is sent from the secondary LU to the primary LU, the secondary send and primary receive values are used. These values control the flow from the secondary to the primary end of the session. The secondary send and primary receive counts are used in the same manner as the primary send and secondary receive counts are when transmitting from the primary logical unit to the secondary logical unit.

The values for the four pacing windows are derived from the following sources:

- The following operands on the MODEENT macroinstruction for the logon mode entry:
 - PSNDPAC for primary send
 - SRCVPAC for secondary receive
 - SSNDPAC for secondary send
- The following operands on the LU definition statement:
 - PACING
 - VPACING
- The following operands on the APPL definition statement:
 - AUTH=NVPACE
 - VPACING.

Note: An application program acting as a primary end of a session can obtain information about pacing counts using the INQUIRE macroinstruction or the LOGON exit routine. For more information, refer to “INQUIRE-Obtain Logical Unit Information or Application Program Status” in *VTAM Programming*.

For secondary to primary pacing, when the application is the primary logical unit, the primary receive count equals the VPACING value on the APPL definition statement, unless the secondary send count is zero (the secondary send count is specified in the logon mode table entry associated with that session). If the secondary send count is set to zero, the primary receive count is also set to zero.

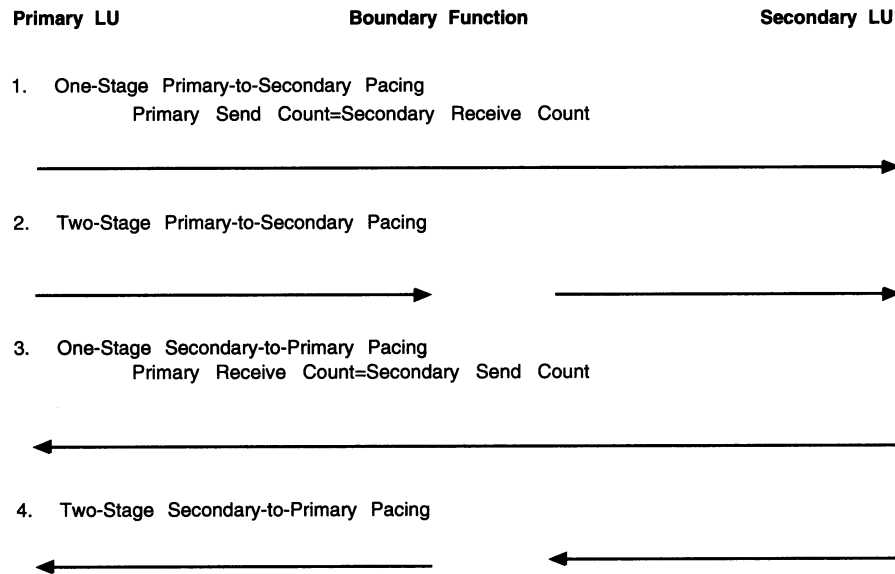


Figure 87. Pacing Counts Used to Control One- and Two-Stage Pacing

For primary to secondary pacing, the secondary receive count is specified with the logon mode table entry unless that value is zero. If zero is specified, the appropriate PACING or VPACING value of the secondary LU is supplied.

The primary send count is indicated in the logon mode table entry. If you specify a nonzero value, then that value is used to build the CINIT. If you specify zero, then VTAM uses the VPACING value specified on the secondary LU definition statement.

Overriding Fixed Pacing Counts

The pacing counts specified on the LU definition statement apply to all sessions in which the LU participates. Values stated in the APPL definition statement override the values in the logon mode table entry and the LU definition statement. When you specify AUTH=NVPACE, the value for the primary send that goes out in the BIND is 0. Therefore, you will have no pacing between the application program and the boundary function.

AUTH=VPACE is normally specified to prevent overloading buffers in the communication controller with outbound messages from VTAM. You can specify NVPACE when the application program sends a limited amount of data and then waits for a user response.

The primary application program can override the primary receive value for a specific session by specifying a nonzero value in the primary receive field of the session parameters it specifies at OPNDST. If 0 is specified, the primary receive value present in the session parameters in the pending logon is used. Only values less than or equal to the value in the pending logon should be specified by the application program.

Note: NVPACE can also be specified when an application program is sending chains of RUs to a secondary logical unit and the application program can not send more BIUs than can be held in the queue. The number of elements in each chain must be no larger than the results of the following formula, where n is the smallest value in the VPACING operands of the definition statements for the LUs:

$$\text{number of elements} \leq 2n - 1$$

If the application program is in session with LUs whose VPACING operands are different from each other, the smallest VPACING values should be used in the formula.

Adaptive Session Pacing

Adaptive session pacing is a dynamic windowing mechanism that is used to pace data transmission on LU-LU sessions. The window sizes of data that can be transmitted are adapted to buffer availability and demand on a session-by-session basis. The session stage endpoints exchange explicit pacing windows that can vary in size during the course of a session. The mechanism thus prevents any single session from monopolizing or exhausting the buffer resources of a session stage endpoint.

Requirements for Adaptive Session Pacing: Adaptive session pacing is used for sessions between VTAM and NCP subarea nodes whenever a programmable device or another application requests the session. For adaptive session pacing to be used the secondary logical unit must be:

- an independent LU
- an application program running in a host that supports adaptive session pacing
- a logical unit controlled by a VTAM and attached to an NCP that supports adaptive session pacing
- a logical unit attached to a VTAM that is performing the boundary function and supports adaptive session pacing.

Adaptive session pacing is used between a peripheral node LU and the subarea node to which it is attached only if the indicator in the BIND specifies that the LU supports adaptive session pacing.

For example, if VTAM is connected to a peripheral node through NCP, there are three session pacing stage endpoints. The NCP and the attached peripheral node, using either fixed or adaptive session pacing, constitute one pair of session pacing stage endpoints. VTAM and NCP, using adaptive session pacing, constitute another pair of session pacing stage endpoints.

When VTAM (rather than the NCP) is providing boundary function for a peripheral logical unit, it always acts as a pacing stage. For example, if there is a session between a VTAM application program and a peripheral logical unit, the following pacing stages exist:

- One (called a subarea stage) between the application program and the boundary function
- One between the boundary function and the logical unit.

Window Size Negotiation: Adaptive session pacing can be used for VTAM's or NCP's boundary function and the attached peripheral node LU. Each session stage endpoint (peripheral node, NCP, and VTAM) can determine the transmission window and include the window size in the pacing response sent to the partner session stage endpoint.

Figure 88 on page 335 illustrates that the session stage endpoint, acting as the transmitter, requests a pacing response from the session stage endpoint that is the receiver when sending the first PIU in the current window. The receiver simply

returns a pacing response that contains the next window size it considers appropriate for the next transmission. If the sender transmits the entire window of PIUs before the pacing window response is received, the sender must wait until the pacing response is received. The sender can request a larger window size by setting an indicator in the next transmission. The receiver can ignore the request or consider the requested window size when it decides the appropriate window size to use next in the response to the pacing request. The sender is obligated not to exceed the window size specified by the receiver.

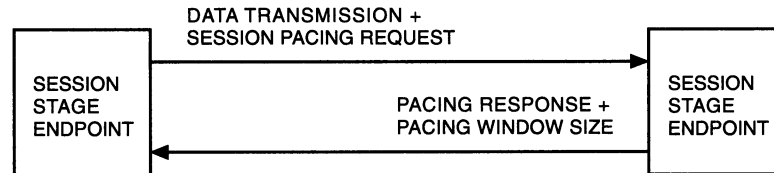


Figure 88. Adaptive Session Pacing Window Mechanism

Adaptive Session Pacing Values: With adaptive session pacing, the window values determined during session setup are either used as tuning values for the actual windows or are ignored. When used for tuning, these values represent the minimum window sizes that are actually used during normal operation.

If the application program is the primary logical unit and the peripheral node is the secondary logical unit and does not support adaptive session pacing, adaptive session pacing is used within the subarea network, and fixed session pacing is used when transmitting to the peripheral node. The boundary node (VTAM or NCP) uses the value of the PACING operand on the secondary logical unit or the secondary receive count (SRCVPAC) in the logon mode entry to pace transmission to the LU. To pace receipt of data from the secondary logical unit, the boundary node (VTAM or NCP) uses the VPACING value on the primary logical unit or the secondary send value (SSNDPAC) in the BIND.

In Figure 89 on page 336, VTAM is performing pacing for the application program and the application program is the primary logical unit. The primary receive count in this case is obtained from the VPACING value on the APPL definition statement.

The following statements describe how VTAM paces traffic destined for a locally-attached peripheral node. Whether the peripheral logical unit is the secondary or primary logical unit determines which BIND pacing counts are used.

- For the subarea stage, if the value of the primary or secondary receive count in the BIND is zero, VTAM's boundary returns the following pacing window sizes:
 - If the session partner is a resource in another domain, the window size is the maximum virtual route pacing window size.
 - If the session partner is a same-domain application program, the window size is 7.
- If VTAM's boundary function is performing pacing to a peripheral node without an NCP and if the value of the primary or secondary receive count is zero, VTAM returns the following pacing window sizes to the peripheral logical unit:
 - If the peripheral node is attached by an SDLC connection, the window size is 7.
 - If the peripheral node is attached by a local-SNA connection, the window size is based on the MAXBFRU specification.

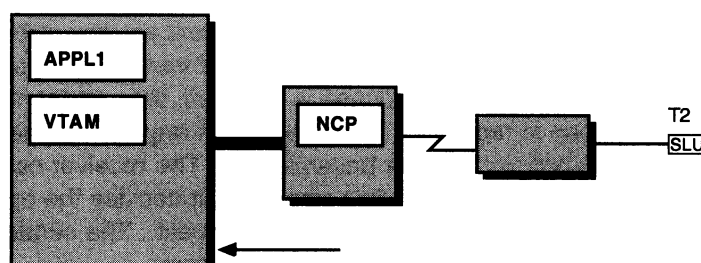


Figure 89. Pacing: Inbound to VTAM When Application Program Is PLU

- If the peripheral node is attached by a token-ring or X.25 connection, the window size is 128.
- The fixed window value that is returned in the pacing response is used for the duration of the session and does not dynamically raise and lower the window size.

Note: The window size value is calculated by NCP using an internal algorithm.

VTAM's Role In Adaptive Session Pacing: An important factor in tuning session pacing is that VTAM, though it supports the adaptive session pacing mechanism, does not dynamically vary its own pacing windows. VTAM always uses the same receive window sizes for adaptive session pacing that it uses for fixed session pacing. VTAM is involved in session pacing in two functional areas: when it is sending and receiving data for an application program and when it is providing boundary function for a peripheral logical unit.

Figure 90 on page 337 illustrates how VTAM returns a pacing response that contains the fixed window size of either of the receive counts. If the value of either of the receive counts is zero, VTAM returns 32767 as a pacing window size.

Figure 91 on page 338 illustrates how a VTAM performing a boundary function for a peripheral node can support adaptive session pacing.

VTAM honors the pacing windows sent by its pacing stage partners (NCP and other VTAMs) for adaptively paced sessions. The next window size used for data transmission is whatever value was received from the NCP in the pacing response.

Independent Logical Units and Adaptive Session Pacing: Although some type 2.1 peripheral nodes do not support adaptive session pacing, an independent logical unit in that node can still act as the primary LU and therefore sends the BIND. The primary pacing values in the BIND are established by the independent logical unit (primary). However, the value of the primary send count might be so large that it can impact the subarea node to which it is attached. For example, a PS/2 attached to an NCP using a token ring (NTRI) can specify a value of zero (no pacing) as the primary send count, which if not altered, can cause the NCP to enter slowdown. VTAM, in conjunction with NCP, allows you to control the primary pacing values used by an independent logical unit for which adaptive session pacing is not supported. If a type 2.1 peripheral node that supports adaptive session pacing is attached directly to VTAM, VTAM honors the pacing window size value included in the pacing response from the peripheral node but continues to use the appropriate receive count for its fixed window size value.

The primary send count specified in the logon mode entry of the primary logical unit is used by VTAM to control the rate at which a peripheral node acting as a primary

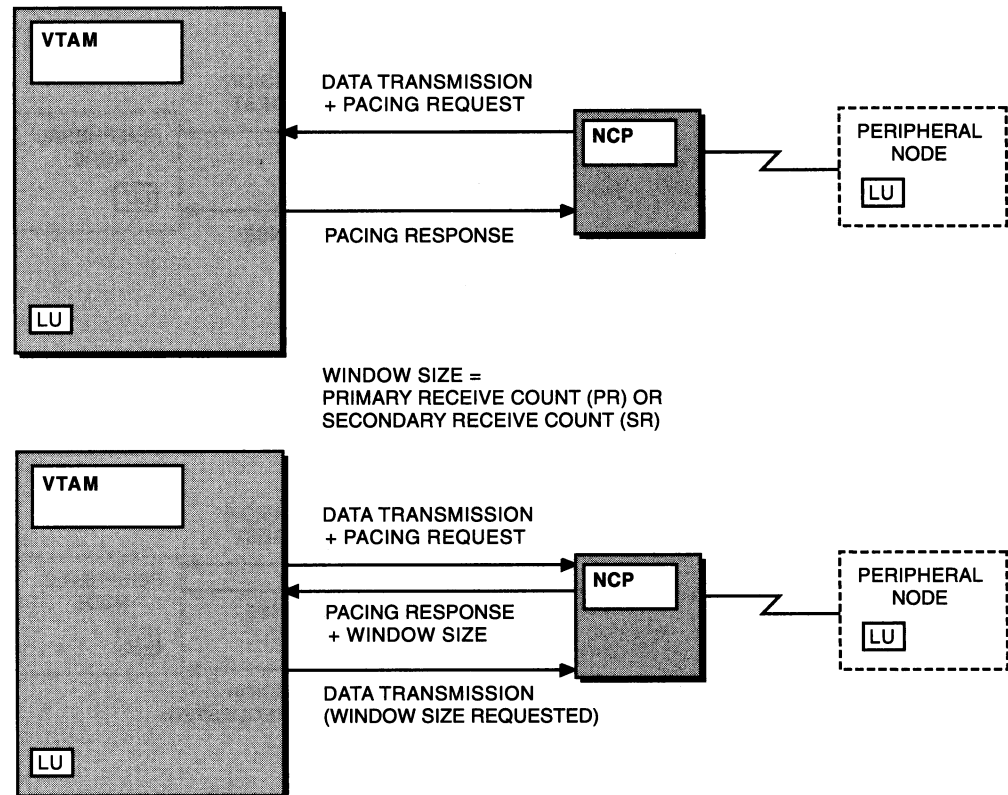


Figure 90. VTAM and NCP Adaptive Session Pacing Window Size

logical unit sends data to an NCP. This value is then used by NCP or VTAM for a directly attached type 2.1 peripheral node to control the primary send pacing. If a negotiable BIND is used, the smaller value (other than zero) is used. If the BIND is non-negotiable and the primary send value in the original BIND (other than zero) is larger than the value VTAM obtained from the logon mode entry or PACING specification, VTAM or NCP may reject the session setup request. For a directly attached type 2.1 peripheral node, VTAM performs the same type of operation.

Therefore, you need to ensure that you specify proper pacing values in VTAM for any type 2.1 independent logical unit. If the type 2.1 peripheral node supports adaptive session pacing, this procedure is not necessary.

Application Program Pacing

This section describes how to use the APPL definition statement to specify pacing for application programs.

For information on virtual route pacing, which affects applications but is not controlled at the application level, see "Virtual Route Pacing" on page 310. For more information about pacing, refer to "Pacing and Flow Control" on page 159.

Inbound Session Pacing

Application programs can act as either primary or secondary LUs. If the application program is the primary logical unit, you define pacing with two values: a nonzero value in SSNDPAC on the secondary LU-mode table entry and the pacing value on the VPACING operand of the APPL definition statement. This way you can pace normal-flow requests from another logical unit.

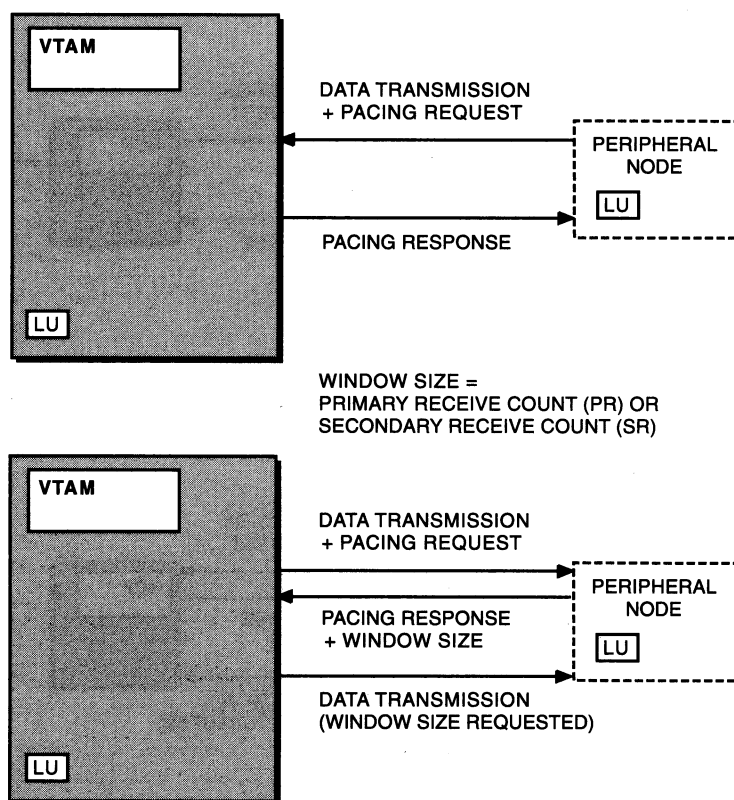


Figure 91. VTAM Adaptive Session Pacing Window Size

If the application program is a secondary logical unit, you define pacing with either value: a nonzero value in PSNDPAC on the secondary LU-mode table entry, or the pacing value on the VPACING operand of the secondary logical unit's APPL definition statement. If you code both values, the nonzero value in PSNDPAC overrides the VPACING value.

Outbound Session Pacing

When the application program is acting as the primary logical unit, you can specify AUTH=NVPACE to nullify pacing from that application program. VTAM does not wait for a pacing response before sending data to the NCP on behalf of this session. VPACE, the default, is normally used to prevent overloading buffers in the communication controller with outbound messages from VTAM. However, you can specify NVPACE when either of the following situations occur:

- The application program sends only single-element request units (RUs) to any one logical unit and, after sending each RU, waits for a response before sending the next RU.
- The application program sends chains of RUs containing a limited number of elements and does either of the following:
 - After sending one chain, waits for a response before it starts sending the next chain
 - Sends the change direction indicator in the last element of the chain.

Local flow control is a part of network flow control and can affect application program design. For primary or secondary application programs, macroinstructions that cause data to be sent on a session (SEND, SESSIONC, OPNDST, or

CLSDST) are not posted complete when a virtual route (VR) is blocked due to congestion. The application program is posted when the VR is again opened. The data to be sent is left in application program buffers.

An application program that issues a synchronous SEND request does not receive control until the request is processed, even if the route is not blocked. For an asynchronous request, the program receives control as soon as the request is accepted by VTAM. However, further processing of the SEND RPLs for this session is suspended as long as the VR is blocked.

Setting Pacing Values

Pacing window values are derived from the following sources:

- Operands on the MODEENT macroinstruction for the logon mode entry
- Operands on the LU and APPL definition statements.

Figure 92 on page 340 and Figure 93 on page 341 show how VTAM selects pacing window values. Depending on the type of SLU, these figures show the following:

- The direction of session pacing
- The number of pacing stages
- How the pacing window values are derived.

Note: For fixed session pacing, the pacing window values derived from these figures are used; however, for adaptive session pacing, these values are either used for tuning or ignored.

The letters used in Figure 92 on page 340 and Figure 93 on page 341 indicate the methods that VTAM uses to determine pacing window values. Table 24 indicates what method corresponds to each letter. Both Figure 92 on page 340 and Figure 93 on page 341 assume that the application program is always the primary logical unit.

If the BIND is an extended BIND, then adaptive session pacing is used. Adaptive session pacing supports only one stage pacing in both directions.

Note: XBSI is the extended BIND indicator in the session initiation request. XBSI=ON means that extended BIND is used to establish the session.

Table 24 (Page 1 of 2). Correspondence of Methods to Letters Used in Following Figures

Letters Used in Following Figures	Methods Used by VTAM
A	IF SRCVPAC is not 0, use it ELSE use SLU VPACING (default is 1)
B	IF SRCVPAC is not 0, use it ELSE use SLU PACING (default is 1)
C	Use SRCVPAC
D	IF SSNDPAC is 0, use it ELSE use PLU VPACING (default is 7) Note: PLU can reduce value, but not to 0.
E	Use SSNDPAC

Table 24 (Page 2 of 2). Correspondence of Methods to Letters Used in Following Figures

Letters Used in Following Figures	Methods Used by VTAM
F	IF PLU AUTH=NVPAGE, use 0 ELSE IF PSNDPAC is not 0, use it ELSE use SLU VPACING (default is 2)
G	IF PLU AUTH=NVPAGE, use 0 ELSE IF PSNDPAC is not 0, use it ELSE use SLU VPACING (default 1)
H	Use PSNDPAC
I	IF SSNDPAC is 0, use it ELSE use PLU VPACING (default is 7) Note: PLU might not reduce value.

	If SNA SLU Is:															
	An Application				An NCP LU				A Same-Domain Local LU				A Cross-Domain Local LU			
And the Pacing Direction Is:	P→S	S→P	P→S	S→P	P→S	S→P	P→S	S→P	P→S	S→P	P→S	S→P	P→S	S→P	P→S	S→P
Then the Number of Stages Is:	1	1	2	1	2	2	2	2	2	2	2	2	2	2	2	2
And the Windows Are:	PS	SR	SS	PR	PS	SR	SS	PR	PS	SR	SS	PR	PS	SR	SS	PR
If an Extended BIND Is Sent, Windows Are Set Using Method:	A	I	D	F	B	I	D	A	I	D	G	B	E	D		
If an Extended BIND Is Not Sent, Windows Are Set Using Method:	A		D	F	B		D	A		D	G	B	E	D		

Figure 92. Pacing Windows for SNA LUs. Coding is from the perspective of the owning SSCP acting as the primary logical unit.

	If Non-SNA SLU Is:					
	A Same-Domain LU or an NCP Terminal		A Cross-Domain 370 CA Terminal		A Cross-Domain Local LU	
And the Pacing Direction Is:	P→S	S→P	P→S	S→P	P→S	S→P
Then the Number of Stages Is:	1	0	2	0	1	1
And the Windows Are:	PS=SR	SS=PR	PS	SR	SS=PR	PS=SR
Windows Are Set Using Method:	C	E	H	C	E	C

Figure 93. Pacing Windows for Non-SNA LUs. Coding is from the perspective of the owning SSCP acting as the primary logical unit.

Operating VTAM

This section provides information to help you make decisions about how to run and control your VTAM domain. You can use this information to write a run book for your operators and to make decisions on setting up any program operators you might use. The following topics are covered in this section:

- Starting the domain
- Activating resources
- Monitoring the domain
- Controlling the domain
- Deactivating resources
- Halting VTAM
- Automatic operations.

Starting the Domain

To start VTAM, use the **VSE EXEC** or **MVS, VM START** command. You can either enter start options when you start VTAM, or you can specify a start list. See “Start Options” on page 127 for more information on start lists and start options.

Following are the general start procedures for VTAM under each operating system:

Start procedures for MVS

1. Respond to message IEA101A during operating system IPL by entering RERP=xx, where xx represents the last two characters of the alternate ERP list's name, if you want to use the resident error recovery procedures (ERP) option or the ERP option with an alternate ERP list.
2. Make available the necessary channel-attached devices using the MVS system VARY command.

3. If using the encryption facility, activate the IBM Programmed Cryptographic Facility licensed program before starting VTAM.
4. Use the VTAM start procedure.
5. Enter VTAM start options, if necessary.
6. Enter the logon manager START command, if necessary.

Start procedures for VM

1. Log on to the VTAM virtual machine, and execute its PROFILE EXEC procedure.
2. Make available the necessary channel-attached devices and communication adapter-attached devices; the devices can also be attached in the PROFILE EXEC.
3. Enter VTAM start options, if necessary.
4. Start the VM/SNA console support (VSCS) component of VTAM, if necessary.

Start procedures for VSE

1. Use the VSE ADD command to make available the communication adapter-attached lines and channel-attached devices needed at start time.
2. Ensure that the necessary VTAM modules are loaded into the shared virtual area (SVA). See "Calculating VTAM Virtual Storage Requirements" on page 107.
3. Use a START procedure to start the partition in which VTAM runs.
4. Enter the JCL statements for starting VTAM, either from the system console or from a device you have assigned for SYSRDR. You can enter the JCL as separate statements from an input device, or you can code the statements in a procedure library and invoke them by entering the EXEC command. For details on filing and using a procedure and for specifying modifications to a stored procedure, see *VSE Operation*.
5. Certain IBM-supplied programs can be attached as VTAM subtasks with the MODIFY SUBTASK command.

Configuration Restart (MVS, VSE)

You can use configuration restart to restart your network with the same status information prior to VTAM's deactivation or failure. VTAM can maintain certain information about VTAM resources, such as status (active or inactive) and operator-specified values of activation and operating parameters. When restarting VTAM after a halt or failure or when reactivating an individual major node after a deactivation or failure, VTAM can use this information to restore the resources to their status prior to the deactivation or failure.

Recording Changes to the Network Configuration: When the host operating system is generated, VTAM definition statements are filed in the VTAM definition library. The first time a major node is activated, activation information is recorded in the resource definition table and in the NODELST file, if one exists. Any further activations or deactivations of major nodes are also recorded in the resource definition table and in the NODELST file (if any). Every time the VTAM operator

changes the status of a minor node, the change is recorded in the resource definition table and in a configuration restart file, if one exists.

The network configuration can be restored to either its initial status or its status before failure or deactivation. The status being restored depends on whether you have defined configuration restart files and on how you use the CONFIG operand and the optional COLD or WARM operand when restarting the network. Figure 94 illustrates this process.

Using Configuration Restart's WARM Start: As shown in Figure 94, if you specify the WARM operand when activating a major node, VTAM updates its resource definition table from information in the VSAM configuration restart file for the major node. VTAM then restores the minor nodes within the major node to their status prior to the failure or deactivation.

Also shown in Figure 94, if you specify the WARM start option when starting VTAM, VTAM can update its resource definition table from information in the NODELST file.

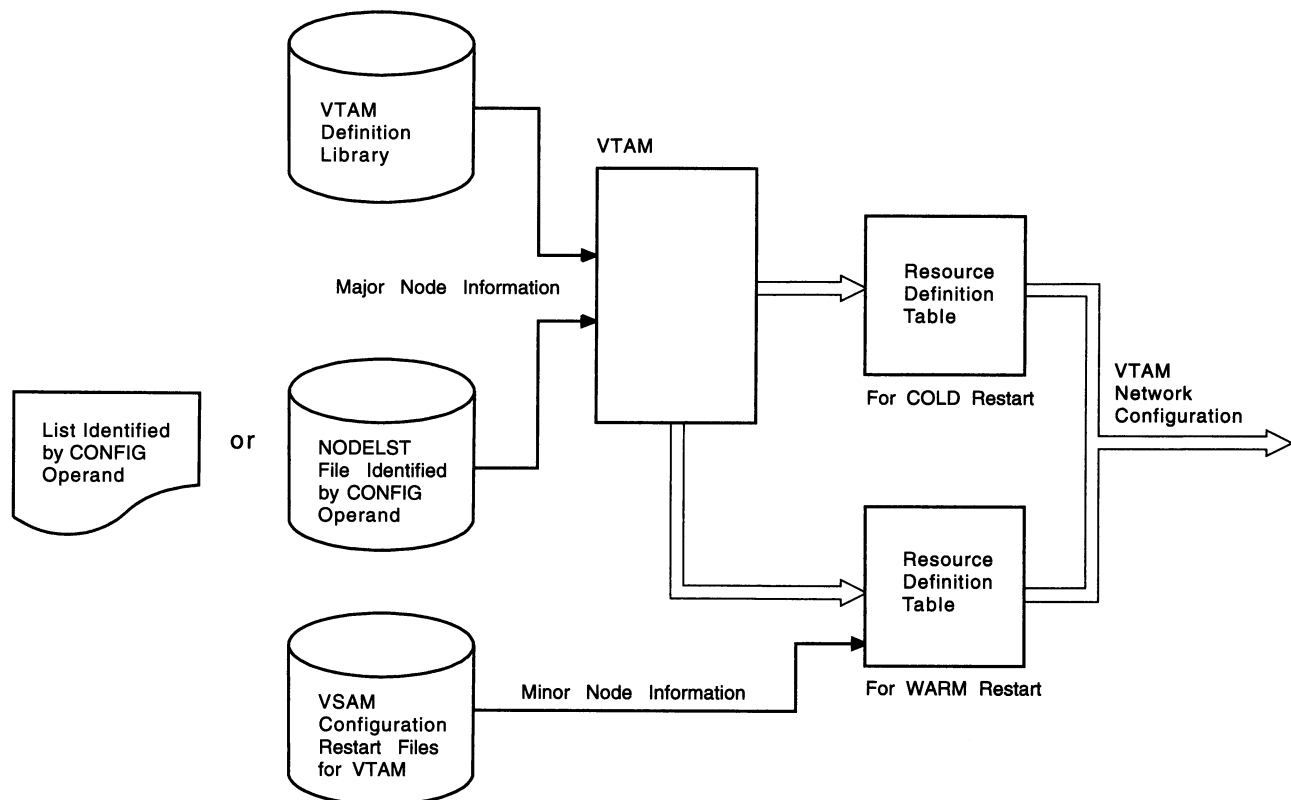


Figure 94. Restoring Resource Definitions with Configuration Restart and NODELST Files

If you specified a NODELST file when the network was last started, the major nodes that were active and the DR files that were applied when the failure occurred can be reactivated by specifying the NODELST file name on the CONFIG operand during a WARM start. If you used the VARY ACT,UPDATE technique of dynamic reconfiguration, major node definitions reflect dynamic reconfiguration changes, and the changes are reestablished when the major node is activated during NODELST processing. See “Dynamic Reconfiguration and Change of Operands” on page 149 for more information on the techniques that are available for dynamic reconfiguration.

Note: Dynamically defined independent logical units cannot be warm-started.

For an example of how NODELST and configuration restart files are used, see Figure 95 on page 345.

In this example, major nodes A, B, and C have configuration restart files (CONFGA for major node A, CONFGB for major node B, and CONFGC for major node C). Each major node is comprised of four minor nodes (for example, major node A is comprised of the minor nodes A1, A2, A3, and A4). When the minor nodes are defined, the initial status (active or inactive, for example) can be specified. In this case, the minor nodes B3 and B4 are to be initially active and the other minor nodes, inactive. This configuration can be represented as follows:

	Major Node A	Major Node B	Major Node C
Minor Nodes	A1 A2 A3 A4	B1 B2 B3 * B4 *	C1 C2 C3 C4

* Defined as initially active

Configuration Restart Files

CONFGA	CONFGB	CONFGC
empty	empty	empty

When VTAM is started, the VTAM operator can enter start options. In this example, the VTAM operator uses the CONFIG operand to specify the list XX, which lists major nodes A and C as initially active. The operator uses the NODELST operand to specify that the VSAM file ABC is to be used to record which major nodes are active in the network:

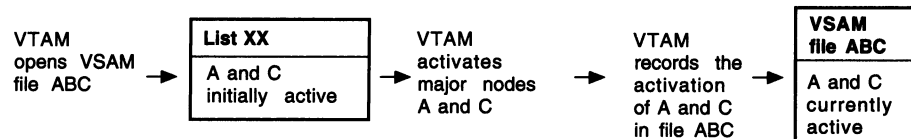
CONFIG=XX,NODELST=ABC

Because of these start options:

VTAM activates major nodes A and C if it is able to open the NODELST file ABC.

VTAM records the activation of major nodes A and C in file ABC.

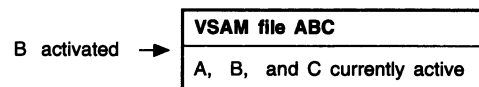
These actions can be represented as:



Then, to activate major node B, the operator enters:

VARY NET,ACT,ID=B

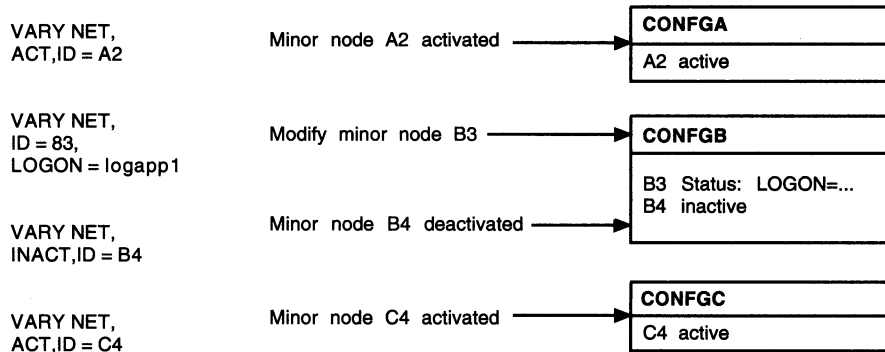
VTAM activates major node B and records its activation in ABC:



VTAM also activates minor nodes B3 and B4 because B3 and B4 are indicated as initially active in the definition statements for major node B. Because B3 and B4 are defined as initially active, VTAM does not record the activation of B3 and B4 in CONFGB.

Figure 95 (Part 1 of 2). Using Configuration Restart and NODELST Files

After the major nodes are active, the operator activates, deactivates, or modifies the status of minor nodes with VARY commands. Every time VTAM activates, deactivates, or modifies the status of a minor node, it records that action in the configuration restart file (if one exists) for the major node of which the minor node is a part.



The status information stored in the configuration restart files can be represented as:

Major Nodes

List XX	File ABC
A and C initially active	A,B, and C currently active

Minor Nodes

Configuration Restart Files		
CONFGA	CONFGB	CONFGC
A2 active	B3 status: LOGON=... B4 inactive	C4 active

If the network is now halted or if a failure occurs in it, VTAM deactivates all the active major and minor nodes and closes the configuration restart files. The information needed to restore the network is in the VTAM definition statements and these files. When the VTAM operator restarts the network, start options can again be entered.

Start option entered by the operator	Major nodes restarted	Minor nodes restarted
Config = XX,COLD	A and C	None
Config = XX,WARM	A and C	A2 and C4
Config = ABC,COLD	A,B, and C	B3 and B4
Config = ABC,WARM	A,B, and C	A2,B3, and C4

Specifying CONFIG=XX,COLD restores the network to its initial status, while CONFIG=ABC,WARM restores the network to the status it had when it failed or was deactivated. Specifying only CONFIG=XX is the same as specifying CONFIG=XX,COLD because if neither COLD nor WARM is coded, COLD is assumed.

The operator can also specify another NODELST file:

CONFIG=ABC,WARM,NODELST=QRS

Or, the operator can continue using the one used before:

CONFIG=ABC,WARM,NODELST=ABC

Figure 95 (Part 2 of 2). Using Configuration Restart and NODELST Files

VTAM uses the VSAM configuration restart files to record changes to the named minor nodes within the major nodes (unnamed minor nodes are not checkpointed). If the VSAM files are defined, the VTAM operator can specify how VTAM is to use the files for configuration restart.

To associate VSAM configuration restart files with individual major nodes, code the CONFGDS and CONFGPW operands on the definition statements for the major nodes.

A NODELST file can be defined to record which major nodes and DR files are active in the network. A NODELST file is specified with the NODELST start option.

You can use either or both of these file types when using configuration restart.

If the user defines configuration restart files, VTAM provides two additional forms of recovery: manual switching to a backup host processor and manual switching to a backup communication controller.

If you do not associate a configuration restart file with a major node, or if the configuration restart file is empty, then the major node is not activated to its prior status.

If NODELST is in effect and you are doing a warm start of a configuration that has been dynamically reconfigured using the VARY DRDS technique, the DR statements that are executed again as part of the start process might fail because the nodes they specify became active earlier in the process.

Information Recorded by Configuration Restart

The following major nodes have the indicated information recorded:

NCP major node

- The channel device address (if channel-attached) and the RNAME values (if link-attached)
- For each line, its status (active or inactive), non-SDLC line scheduling parameters, and for dial SDLC lines, the answer mode (whether the line is dial-in)
- For each link station, its status (active or inactive)
- For each logical unit, its status (active or inactive) and the names specified in the LOGAPPL and LOGMODE operands
- For each physical unit, its status (active or inactive)
- For each logical unit, the cryptographic status (REQD, SEL, OPT, or NONE).

Channel-attachment major node

- For each link, its status (active or inactive) and the channel unit address (CUA)
- For each link station, its status (active or inactive).

Local SNA major node

- For each physical unit, its status (active or inactive) and its channel device address
- For each logical unit, its status (active or inactive) and the names specified in the LOGAPPL and LOGMODE operands and whether it is active or inactive

- For each logical unit, the cryptographic status (REQD, SEL, OPT, or NONE).

Switched SNA major node

- For each logical unit, its status (active or inactive) and the names specified in the LOGAPPL and LOGMODE operands
- For each physical unit, its status (active or inactive)
- For a physical unit with dial-out capability, the PATH=USEINOUSE setting
- For each logical unit, the cryptographic status (REQD, SEL, OPT, or NONE).

Local non-SNA major node: Each channel-attached terminal, its status (active or inactive), its channel device address, and the names specified in the LOGAPPL and LOGMODE operands.

CDRSC or CDRM major node: The status of each minor node (active or inactive).

Activating Resources

You can initially activate resources by doing one of the following:

Specifying the name of the resource in a configuration file

This way, the resource is activated when VTAM is initialized. For information on coding a configuration file, see "Configuration Lists" on page 134.

Using the VARY ACT command after VTAM is initialized

Enter a VARY ACT command with the ID operand specifying the name of the node to be activated.

To control activation of resources that are lower in the hierarchy, you can do one or both of the following:

Use the SCOPE operand on the VARY ACT command

If you code or use the default SCOPE=COMP on the VARY ACT command, all resources subordinate to this resource are activated if the ISTATUS operands for each are defaulted or coded as active. SCOPE=ALL causes all resources to be activated regardless of the coding of the ISTATUS operand.

Code the ISTATUS operand on the resource's definition statement

The ISTATUS operand controls the initial status of the resource. If you code or default to ISTATUS=ACTIVE, this resource is activated when its major node is activated.

If you make changes to your definition statements, deactivate and then reactivate the major nodes that are affected. Otherwise, the changes are not recorded.

Order of Activation

A resource cannot be activated unless all other resources above it in the hierarchy have been activated. For example, a logical unit cannot be activated unless the physical unit to which it is subordinate is already active. When a resource is deactivated, all of the resources subordinate to it in the hierarchy are also deactivated.

Although there is no logical hierarchy within the set of major nodes themselves or between resources in different major nodes, there is an obvious topological relation-

ship among the major nodes representing subarea nodes in the network and the lines and link stations connecting them. Therefore, the activation of an NCP major node depends on the lines, link stations, and other subarea nodes that constitute the routes between VTAM and the NCP being active. Likewise, deactivation of an NCP major node or a line or link station within it can affect those other subarea nodes for which the resource being deactivated constitutes a part of the path from VTAM.

All of the elements of a communication path between two session endpoints must be active before a session can be established. In domains with more than one subarea, a path definition set must be activated before communication between subareas can be attempted. Major nodes are activated, and then their minor nodes are activated to enable sessions to be established for carrying user message traffic.

Although an activation command for a major node can be entered at any time after VTAM is initialized and before it is halted, the activation of NCP major nodes and external CDRM minor nodes requires fully operative explicit routes. If these routes are not available, the activations remain pending until the routes become available or until you deactivate the resources.

If automatic logons to a controlling application program have been defined for any logical units, all of the necessary application program major nodes must be activated before any of these logical units are activated (directly or indirectly). The logical units must be active before a session can begin.

The specific order in which you activate major nodes depends on how you have coded your network. For example, if you defined a terminal to automatically logon to an application program, activate the application program first. Following is a general guideline for activating your resources (the activation order may vary due to coding differences in your network):

1. Local resources — the host CDRM, application programs, and other local resources. However, avoid activating any resources that will automatically logon to resources in other domains.
2. Routes — explicit and virtual route path definitions. Activate these paths before any connection to another subarea node (NCP or VTAM).
3. Peripheral resources — channels, NCPs, switched devices, cross-domain resource managers and cross-domain resources.

Activating Application Programs

Before an application program can use VTAM services, the application program minor node that defines the application program must be activated. An application program minor node is typically activated at VTAM startup by placing the name of the application program major node that contains the application program minor node into the VTAM configuration list.

After an application program minor node is activated, it is up to the application program to open its ACB and begin using VTAM services. The order in which the minor node is activated and the application program is started is not critical, but the minor node must be active when the application program attempts to open its ACB for VTAM.

If the operator cancels an application program, the operating system forces the application program's disconnection from VTAM. For the operator to be able to

cancel an application program that is active to VTAM, the operator must first associate the application program's defined VTAM name with:

- **MVS, VSE** The application program's job name
- **VM** The user ID of the application program's virtual machine.

The logon manager, which runs as a VTAM application, can be terminated with the MODIFY STOP command. For information on using the MODIFY STOP command, see "MODIFY STOP Command" in *VTAM Operation*.

VM The application program is started by the following process:

1. Log on the virtual machine in which it runs
2. Execute an IPL command for the GCS group containing VTAM
3. Use either the OSRUN command of GCS or the LOADCMD command of GCS
4. Enter the appropriate program command.

If the application program runs as a subtask of VTAM, you must use the MODIFY SUBTASK command to start your application program.

Monitoring the Domain

VTAM automatically provides you with messages that contain information about the VTAM domain. If more information is required, various DISPLAY commands are available to request status information about domain resources, and to verify changes resulting from previous operator requests.

Using the DISPLAY Command

In general, DISPLAY commands can be used to obtain status information about active major nodes and their minor nodes (active or inactive). Some DISPLAY commands can be used to obtain information about resources that are not nodes. For example, the DISPLAY BFRUSE command provides information about VTAM's use of buffers, and the DISPLAY COS command provides information about a network's class-of-service tables.

Monitoring I/O Problems

You can change how long some VTAM I/O operations and internal procedures remain pending before VTAM notifies you. To do this, you use the IOINT start option, the MODIFY,VTAMOPTS command, or the MODIFY,IOPD command to change the I/O problem determination time-out interval. For example, if you use the command MODIFY,IOPD,IOINT=120, VTAM notifies you of I/O operations or internal procedures that remain pending for 120 seconds or more. After you are notified of a pending I/O operation, you can take any necessary action.

If you do not want to be notified of pending operations, you can specify IOINT=0. This deactivates the I/O problem determination function.

For information about the IOINT start option, see "IOINT" in the *VTAM Resource Definition Reference*. For information about the MODIFY,VTAMOPTS and the MODIFY,IOPD commands, see "MODIFY VTAMOPTS Command" and "MODIFY IOPD Command" in *VTAM Operation*. See "Modify Input/Output Problem Determination" in *VTAM Diagnosis* for more information about the IOPD trace facility.

Suppressing Messages

To help you control the amount of information you receive when monitoring the network, VTAM allows some messages to be suppressed. Messages are divided into the following categories based on suppression level (in order, from the lowest suppression level to the highest):

1. Informational
2. Warning
3. Normal
4. Serious
5. Insuppressible.

The categories to be suppressed can be specified, either with the SUPP start option or with the MODIFY SUPP command. By specifying one category, VTAM suppresses all messages in that category and all messages in lower categories. For example, if the operator issues MODIFY SUPP=NORM or if you code SUPP=NORM in the start options, all messages within the normal category and those below it, warning and informational, are suppressed.

Note: You cannot suppress messages classified as “insuppressible.” Examples of insuppressible messages are those that solicit information from the operator, and messages that are responses to DISPLAY commands.

The MODIFY SUPP command changes only the categories of messages that are suppressed, not the suppression level of messages. You can change the suppression level of a message by using USS facilities.

For information on the suppression level of each VTAM message, see “Message Suppression Levels” in *VTAM Messages and Codes*.

Message Flooding Prevention

VTAM messages that could potentially flood the operator console can be suppressed by a message flooding prevention facility. This facility recognizes and filters duplicates occurring within a given time span, thus preventing critical information from being buried in a mass of repetitious information.

You can choose which messages are candidates for message flooding prevention by copying, renaming, and modifying VTAM's default table (ISTMSFLD) to create your own message-flooding prevention table. Suppression criteria (variable text similarity) and suppression time intervals may be specified for each message in your table, and messages selected for suppression can also be suppressed from the hardcopy log. Message-flooding prevention support also provides the ability to modify a message flooding table dynamically.

For complete information on how to define a message-flooding prevention table, see “Message-Flooding Prevention Table” in the *VTAM Resource Definition Reference*. For information on modifying (MODIFY TABLE) and displaying (DISPLAY TABLE) the message-flooding prevention table, see *VTAM Operation*.

Some VTAM messages can also be controlled using the following VTAM start options. For information on the ASIRFMSG, ESIRFMSG, FSIRFMSG, SIRFMSG, IOMSGLIM, PLUALMSG, and SLUALMSG start options, see Chapter 4, “Start Options” in the *VTAM Resource Definition Reference*.

Using the Online Message Facility (MVS, VM)

The online message facility is an OS/2 program that provides online access to information from *VTAM Messages and Codes*. The facility helps network operators and system programmers operate and diagnose VTAM without interrupting those tasks.

You can retrieve the descriptions of host messages by clicking on the messages. If a message you select contains certain types of system codes (such as sense codes and abend codes), the program includes the descriptions of those codes with the description of the message. You can use the facility to retrieve message and code information without connecting to a host.

For more information about the Online Message Facility, see *Using the Online Message Facility*, available on the *IBM Networking Softcopy Collection Kit CD-ROM*.

Displaying and Testing Routes

The DISPLAY ROUTE command provides information about what routes are available between the VTAM host node and a destination subarea node. In addition, the TEST operand of the DISPLAY ROUTE command allows the operator to test the explicit routes between the host node and a destination subarea node for the routes' ability to transfer data between the subareas. The routes to be displayed can be specified by explicit route number, virtual route number, or class-of-service name. If the TEST operand is specified and the routes to be displayed are specified by virtual route number, the explicit routes tested are those to which the specified virtual routes are currently mapped. If the TEST operand is specified and the routes to be displayed are specified by class-of-service name, the explicit routes tested are those to which the virtual routes in the specified class of service are currently mapped.

You can also test explicit routes starting in an NCP. The ORIGIN operand on the DISPLAY ROUTE command specifies the NCP in which the route starts. Although the originating node must be NCP Version 3 or later, the subarea at the other end can be any of the products that can coexist with VTAM Version 3 or subsequent releases.

Defining Operator Messages and Commands

You can modify certain messages and operator commands. These commands and messages are defined in unformatted system services (USS) tables and are called USS messages and USS commands. Whenever VTAM receives a USS command, it uses either an IBM-supplied default USS table or a user-written USS table to process it. Similarly, whenever VTAM is to send a USS message, it uses one of these tables to determine the message level (the message associated with this release of VTAM or a previous release), message text, and other characteristics of the message. The IBM-supplied operation-level USS table is named ISTINCNO.

By creating a supplementary USS table, you can change the text or other characteristics of a message or the syntax or default values for a command. You can create supplementary tables using USS macroinstructions to redefine the VTAM commands or messages that you want to change. To use your USS table, specify the name of the table on the USSTAB start option. If you specify a user-defined table for operator commands, the ISTINCNO table is not used. If you specify a user-defined table for operator messages, the ISTINCNO table is used in conjunction with the user-defined USS table. For more information on USS tables, see

“Unformatted System Services (USS) Tables” in the *VTAM Resource Definition Reference*.

Note: Because program operators depend on VTAM messages, changes to the operator messages could disrupt the functioning of a program operator. If you change an operator message, notify the responsible system programmer.

You can specify an operation-level USS table for a program operator using the SSCPFM and USSTAB operands of the program operator's APPL definition statement.

Multiple Console Support (MCS) in VTAM (MVS, VSE)

MVS An operator can enter VTAM commands and receive messages at the system master console and at designated secondary consoles. To enter VTAM commands at a secondary console, the console must be authorized to accept commands from command group 1 (system control group) and command group 2 (I/O control group).

VSE An operator can enter VTAM commands and receive messages at the system console or any other console with master authority.

MVS, VSE A VTAM command response goes to the console where the command was entered. Depending on their contents, VTAM messages use routing codes 1, 2, 4, 6, 8, and 10.

MVS Therefore, a secondary console should be assigned routing codes based on the kinds of VTAM messages that are to be routed to that console.

VSE VSE master consoles are configured to receive all these codes. Using MCS macros, you can implement other consoles and specify a subset of these routing codes based on the kinds of VTAM messages that are to be routed to those consoles.

MVS, VSE To determine the routing codes used by particular messages, see “Message Routing Codes” in *VTAM Messages and Codes*.

MVS The operator can change command authorization and message routing using the MVS command VARY. The following command, for example, designates a secondary console where any VTAM command can be entered and where any VTAM message that has not been suppressed can be received:

```
VARY 01F,CONSOLE,AUTH=(SYS,IO)
```

The channel device name of the console is 01F. It is authorized to enter commands from Command Group 1 (SYS) and Command Group 2 (IO). This console receives all messages that have not been suppressed.

MVS Details about using the VARY command to designate a secondary console are contained in the MVS reference manual *MVS System Commands*.

Controlling the Domain

Part of operating VTAM is the task of starting and stopping sessions under special conditions, controlling incoming and outgoing calls, and replacing or reassociating tables that affect how VTAM operates.

Establishing and Terminating Sessions with Operator Commands

A session is usually established by an LU requesting session establishment. However, you can establish a particular session or set of sessions using the VARY LOGON command, or you can establish a single session between a user and an application program using the LOGON operand of the VARY ACT command. If the logical unit you are activating is defined with a LOGAPPL on its LU definition statement, you can establish a session without using the LOGON operand.

A session is usually terminated by one of the session partners requesting termination. However, you can terminate a particular session or set of sessions without deactivating either of the session partners using the VARY TERM command. You can also terminate a session using the VARY INACT command for one of the logical units participating in the session. This command with a TYPE=IMMED or TYPE=FORCE operand breaks any sessions with the logical unit. (Note that TYPE=IMMED has the same effect as TYPE=UNCOND.)

VM The VSCS FORCE command can be used to force termination of a session with an LU. This command should be used only when the CP FORCE and VARY INACT commands do not work.

Controlling X.25 Connections (VM, VSE)

Switched lines with dial-in capability as defined under a packet major node (LINE statement with CALL=IN or CALL=INOUT) can allow or disallow incoming calls. Allowing or disallowing calls can be changed with the VARY ANS command. A line can be activated in answer state with the ANS operand in the VARY ACT command.

In an X.25 environment, there are no requests for manual dial-out operations. The called network address of the remote physical unit is placed on the PATH definition statement under the switched major node. That path can be made usable or unusable with the VARY PATH command.

Dynamic Table Replacement

The MODIFY TABLE operator command allows you to refresh, replace, or reassociate VTAM tables without halting and restarting VTAM. The following tables are affected by the MODIFY TABLE operator command:

- Associated LU tables
- Class-of-service (COS) tables
- Interpret tables
- Logon mode tables
- Message-flooding prevention tables
- Model name tables
- Session awareness data filter tables
- USS tables.
- **VSE** X.21 call progress signal (CPS) retry table.

Before you can use the MODIFY TABLE command to replace a USS or interpret table, the new table must be assembled using the VTAM macro libraries.

Deactivating Resources

This section provides an overview to deactivating VTAM resources with a VARY INACT command or a VARY REL command. For additional information on deactivating an NCP, see “Deactivating an NCP” on page 381.

Order of Deactivation

VTAM deactivates resources in the opposite order from activation. When VTAM receives a command to deactivate a resource, it first deactivates any resources subordinate to the named resource, moving from the bottom of the hierarchy upward, and then deactivates the named resource. Deactivating resources can be disruptive to existing LU-LU sessions. However, depending on the type of connection, a resource can be deactivated without disrupting the LU session if you use TYPE=GIVEBACK on the VARY INACT or the VARY REL command.

Deactivation of certain VTAM resources requires more attention than does their activation. Be aware of the changing connectivity of the domain and the effects on the rest of the network of NCP major node and channel-attachment major node deactivation. Also consider the effects of cross-subarea link and link station deactivations on the operation of the rest of the network. In particular, you should be aware of the impact of deactivating any of the following:

- An NCP through which other NCPs are attached to the network
- A link or link station, whether channel or SDLC, that provides the only connection between a subarea and the rest of the network.

Automatic Deactivation

A VARY INACT command for an NCP major node can result in the automatic deactivation of cross-subarea links and link stations in adjacent subarea nodes (other NCPs or VTAM itself). For example, if the operator enters a command to deactivate an NCP, VTAM automatically deactivates any links and link stations that were automatically activated in adjacent nodes, provided these adjacent links and link stations were not also directly or indirectly activated.

VTAM does not automatically deactivate a multipoint subarea link, even if all physical units on the line have been deactivated.

Normal Deactivation

Normal deactivation is the default for the VARY INACT command. When normal deactivation is requested, resources are not actually deactivated by VTAM until LU-LU sessions associated with the resources have been terminated. Active and pending-active sessions are not affected by normal deactivation of a resource. All queued requests for sessions involving the resources to be deactivated are discarded and the initiators are notified. No new requests for sessions with these resources are accepted.

Immediate Deactivation

Immediate deactivation is specified with the TYPE=IMMED operand of the VARY INACT command. When immediate deactivation is requested, LU-LU sessions associated with the resources are effectively broken. All queued requests for sessions involving the resources to be deactivated are discarded, and the initiators are notified. No new requests for sessions involving these resources are accepted. All input and output operations for active LU-LU sessions are immediately halted, with possible loss of data. If the deactivated logical unit is in session with an application program, the application program is notified of the deactivation. The application program must complete the session termination for the deactivation to be completed.

Immediate deactivation provides tight control over the domain; normal deactivation provides less stringent control, but allows for a more orderly deactivation. Normal and immediate deactivations are not completed until the proper SNA requests and responses are exchanged between the VTAM SSCP and the affected resources. If conditions in the domain prevent the sending of the proper requests or responses, normal and immediate deactivation might not be successful.

Forced Deactivation

Forced deactivation is specified with the TYPE=FORCE operand of the VARY INACT command. When forced deactivation is requested, all of the actions described for immediate deactivation apply. An application program notified of the termination of a session with a deactivated logical unit might have to complete the session termination (depending on which user exits have been coded for the program) for the deactivation to be completed. Forced deactivation can be used when immediate and normal deactivation are not successful.

VM Use the VARY INACT or the CP FORCE command, respectively, to force the termination of a CP connection and the associated VTAM LU session. The VSCS FORCE command can be used if VARY INACT and CP FORCE are not able to terminate the session.

Forced Reactivation

Forced reactivation is specified with the TYPE=REACT operand of the VARY INACT command. When forced reactivation is requested, VTAM performs a forced deactivation and then attempts to reactivate the resource. However, for lines, PUs, and LUs, VTAM does wait for responses from the resource. Forced reactivation of an NCP reactivates the SSCP-PU session, but has no effect on lines and link stations in adjacent subarea nodes. (In this case, the forced reactivation command should be entered for the adjacent line or link station if one of these resources is the source of the problem.) Forced reactivation does not apply to all types of resources.

Halting VTAM

You can halt VTAM using the HALT or HALT QUICK commands. When VTAM is halted, all of its resources are deactivated, and all sessions involving those resources are terminated. Additionally, if the VTAM being halted is an intermediate routing node, any sessions passing through that VTAM are disrupted. Sessions running under the IBM Network Routing Facility program might be disrupted if the VTAM host is halted. This is true in any situation where the session ends are not in the host subarea. If the HALT or HALT QUICK commands do not execute properly, the operator can cancel VTAM.

The CDLINK operand for the HALT command is the same as for the VARY INACT command, except that the cross-subarea links to which it applies are determined when the HALT command is entered and are not redefined as the domain contracts. That is, the CDLINK operand applies to all cross-subarea SDLC links between NCPs that are cross-domain at the time the HALT command is entered.

Except for channel links, all other cross-subarea links (those that are entirely within the domain at the time the HALT command is entered) remain active during and after the halt, regardless of how CDLINK is specified. That is, their final status does not depend on whether the links and their link stations were automatically, directly, or indirectly activated. This allows sessions using that link to remain active. It also allows nondisruptive, simultaneous NCP deactivations, regardless of how the communication controllers are connected within the domain. Cross-subarea channel links are always deactivated during a VTAM halt.

The HALT command notifies application programs within the VTAM subarea of the domain shutdown and waits for them to close their ACBs. Intermediate routing node traffic passing through the VTAM being halted will be disrupted without notification.

New sessions are not permitted and new ACBs cannot be opened for any resources of the host that have received a HALT command. Except for this restriction, application programs of the host that have received a HALT command can continue their current operations. Any pending-active sessions are terminated, but active sessions remain active. After all application programs have stopped using VTAM services (that is, after they have closed their ACBs), processing is the same as for the HALT QUICK command.

During a normal halt, you can monitor the shutdown of the domain by displaying the status of application programs and other logical units. The VARY TERM command can be used to terminate sessions, and, if necessary, to speed the processing of the HALT command.

The HALT QUICK command can also be used at this time, if necessary, to speed halt processing. The HALT QUICK command notifies application programs of the domain shutdown and then proceeds with the equivalent of the VARY INACT,TYPE=IMMED command for each major node. All active LU-LU sessions are disrupted. After VTAM has received a HALT QUICK command, the only VTAM commands that can be used are DISPLAY commands, VARY TERM commands (to terminate sessions), and VARY INACT,TYPE=FORCE commands (to force the deactivation of resources).

VM,VSE You can also use the MODIFY SUBTASK command to detach application programs running as VTAM subtasks.

If, during HALT QUICK processing, after all the major nodes are inactive, any application programs have not yet closed their ACBs, VTAM displays a message listing the names of these application programs. Because VTAM cannot shut down the domain until all application programs have been disconnected from VTAM, you might want to speed up the halt process by canceling the jobs of application programs with open ACBs. The HALT QUICK command can prevent an application program from using VTAM services, but VTAM cannot force the program to disconnect itself from VTAM. When the operator cancels an application program, the host operating system disconnects it from VTAM.

Canceling VTAM

It might be necessary to cancel VTAM if VTAM cannot be halted in any other way. VTAM should be canceled only if the HALT and HALT QUICK commands do not work properly. Canceling VTAM causes the immediate abnormal termination of VTAM without an orderly shutdown of the domain.

The HALT CANCEL command cancels VTAM. This command depends only on the proper functioning of the host operating system's abnormal termination facilities. The HALT CANCEL command causes an abnormal termination with one of the following system completion codes:

- **MVS** X'0A9'
- **VM** X'13E'.

When VTAM is canceled, no further I/O operations occur. Application programs using VTAM are notified of a VTAM shutdown through their TPEND exit routines, if possible. Data being transmitted on sessions might be lost. If TPEND cannot be scheduled for an application program, the application program is also abnormally terminated, if possible. In some circumstances, you might have to cancel some application programs.

VM VTAM is canceled by using VM's HX, IPL, RESTART, RESET, or LOGOFF commands to stop VTAM's execution in its virtual machine. VSCS also stops if it is running in the same virtual machine. You can also use these commands to stop only VSCS if it is executing within a separate virtual machine.

VSE You can cancel VTAM by canceling VTAM's partition.

Automatic Operations

You can use special application programs to assist the VTAM operator in controlling a VTAM domain. These application programs are either program operators or communication network management (CNM) application programs.

Program Operators

A program operator is an application program that is authorized to issue certain VTAM commands and receive VTAM messages. A program operator can be used to:

- Enter VTAM commands (except for commands to start or halt VTAM) from a terminal in the domain
- Monitor and control elements in the domain at program execution speed
- Define specialized commands (for example, to display the status of the entire domain)
- Reformat responses to VTAM commands (such as to reformat the status display of the entire domain to fit on a 3270 display screen)

Secondary and Primary Program Operators: A program operator can have varying degrees of network management capability, depending on the options specified in its APPL definition statement. If you code AUTH=SPO (secondary program operator) or PPO (primary program operator), the application program can issue VTAM operator commands (except START and HALT) using the SENDCMD and RCVCMDC macroinstructions. The application program can then monitor and control the VTAM domain.

An application program that you specify as a primary program operator (PPO) receives all unsolicited messages, that is, all informational and error messages that are not replies to operator commands. If no application program that is designated as PPO is active when these messages occur, they are directed to the system console.

An application program that you specify as a secondary program operator (SPO) can receive solicited messages, that is, messages that are in reply to VTAM operator commands issued by that program operator only. You can, however, limit the number of messages that can be queued for a program operator by using the POAQLIM operand on the APPL definition statement.

CNM Application Programs

A communication network management (CNM) application program can request and receive information that might be useful in managing an SNA network. The application program can communicate with certain physical units in the network to obtain status information. Appendix K, "Program Operator Coding Requirements" in *VTAM Programming* describes in detail the macroinstructions used to create a program operator.

CNM Routing Table: VTAM refers to a communication network management (CNM) routing table to determine which CNM application program is to receive an unsolicited network services request unit (NSRU) requiring further processing. An application program can put its own procedure-related identifier (PRID) in each request sent to VTAM. When a reply to the request is returned, VTAM uses the PRID to route the reply to the application program. However, unsolicited request units contain network information, but they do not have a PRID. Therefore, VTAM needs the CNM routing table to determine which application program is to receive the unsolicited request unit.

VTAM provides default routing information for IBM CNM licensed programs, such as the NetView program. If you authorize a user-written application program to use the CNM interface, you should write a supplementary CNM routing table to route unsolicited requests to that application program.

For information on the request units for which VTAM provides default routing, see "CNM Routing Table (MVS)" in *VTAM Customization*.

NetView Program and Other CNM Application Programs: The NetView program can be used as a CNM application program or as a program operator. The hardware monitor and session monitor, which are components of the NetView program, can also be used as CNM application programs. Because the NetView program performs the functions of a program operator, code the program operator and CNM interface as authorized (AUTH operand on the APPL definition statement).

AUTH=CNM must be coded for the following application programs:

- NetView session monitor
- NetView alias name translation facility
- Network Logical Data Manager (NLDM) licensed program.

Defining a CNM Application Program: To define a CNM application program, code AUTH=CNM on the APPL definition statement. The application program is then authorized to use the CNM interface to communicate with physical units in the

domain, using embedded network services request units. Such an application program can use the CNM interface to gather error or link test information.

CNM Routing: VTAM refers to a CNM routing table to determine which CNM application program is to receive an unsolicited network-services request unit that requires further processing.

With solicited request units, the application program can embed its own procedure-related identifier (PRID) in each request sent to VTAM. When the reply to the request is returned, VTAM uses the PRID to route the reply back to the application program. Unsolicited request units, on the other hand, are not responses to previous requests, so they do not have PRIDs. Therefore, a CNM routing table is needed to route unsolicited request units to the appropriate CNM application program.

VTAM provides a default CNM routing table called ISTMGC01 for IBM CNM application programs. If you authorize one or more user-written application programs to use the CNM interface, you should write a supplementary CNM routing table to allow routing of unsolicited requests to the appropriate user-written application program. For information on coding a supplementary CNM routing table, see “CNM Routing Table (MVS)” in *VTAM Customization*.

Reducing Session Awareness (SAW) Data

Session awareness (SAW) data includes all information about sessions that is known by VTAM. You might need to reduce the amount of session awareness data passed to a communication network management (CNM) application program. You can do this by filtering the session awareness data. A SAW data filter specifies which application programs receive SAW data and which do not receive data. It can also control from which logical units the CNM application programs receive data.

Using the Default SAW Data Filter: The default session awareness data filter table allows data for all sessions to be passed across the CNM interface. You use the default filter only to reduce the amount of data that is passed to the session monitor, and therefore increase the amount of data that is sent to the CNM application programs.

ISTMGC10 in VTAMLIB contains this default table. You can use the default table, modify the default table or dynamically change it using the MODIFY TABLE operator command, or you can replace it with one of your own.

NetView customers can use the default filter with the session monitor filter. The default filter then supplements the use of the NetView filter. The session monitor filter controls the actual processing of NetView session awareness data and is still needed.

Coding Your Own SAW Data Filter: If you want more control over the amount of data that is filtered, you can use your own SAW data filter table, or you can modify the default table or dynamically change it using the MODIFY TABLE operator command.

You define a session awareness data filter to VTAM using the KEEPMEM, KCLASS, and MAPSESS macroinstructions. KCLASS and MAPSESS are used to define the session monitor filter. NetView customers can copy their existing filter

definitions, modify them slightly, and create equivalent VTAM filters. The only change necessary is to place a `KEEPMEM START` macro at the beginning of the filter definitions and add a `KEEPMEM STOP` macro and an `END` statement at the end of the existing definitions. VTAM ignores any NetView operands that do not apply to the VTAM filter.

Following is a simple example of a session awareness data filter table:

```
ISTMGC10  KEEPMEM  START
SAW       KCLASS   SAW=YES
NOSAW     KCLASS   SAW=NO
M1        MAPSESS  KCLASS=SAW,PRI=SSCP1,SEC=*
M2        MAPSESS  KCLASS=SAW,PRI=IMS,SEC=*,PRINET=NETA
M3        MAPSESS  KCLASS=NOSAW,PRI=*,SEC=T3?7*,PRINET=*,SECNET=NETB
                KEEPMEM  STOP
                END
```

The `KCLASS` macroinstruction specifies whether to pass session awareness data over the CNM interface (`SAW=YES/NO`). The labels given to the `KCLASS` macroinstructions are specified on the `MAPSESS` macroinstruction. `MAPSESS` specifies a PLU-SLU session pair and specifies the `KCLASS` definition that VTAM uses to determine whether to pass session awareness data over the CNM interface.

The PLU and SLU are derived from the `PRI` and `PRINET` operands, and the `SEC` and `SECNET` operands respectively. Each has a maximum length of 8 characters. The special characters `?` and `*` used in the pattern have the following meanings:

- `?` matches any single letter or digit.
- `*` matches any letters or digits and can be used only as the last character in the pattern.

The order of macroinstructions is significant. VTAM examines the `MAPSESS` definitions from top to bottom when searching for a match for the PLU-SLU pair. The macroinstructions in the simple example of a session awareness data filter table cause VTAM to:

- Pass session awareness data to the NetView session monitor for a session with a PLU named `SSCP1` or `IMS`
- Ignore session awareness data for a session with an SLU whose name matches the `T3?7*` pattern if the PLU is not named `SSCP1` or `IMS`
- Pass session awareness data for a session with `IMS` as the PLU, even if the SLU name matches the `T3?7*` pattern, because the `M2` definition is before the `M3` definition.

Operating with an NCP

This section helps you plan to run and control a domain that includes NCP sub-areas.

The following topics are covered in this section:

- Loading an NCP
- Activating an NCP
- Monitoring an NCP
- Deactivating an NCP

- Dumping an NCP.

Loading an NCP

Note: You cannot load or activate an NCP over a type 2.1 link.

VM To load an NCP into its communication controller from VTAM:

1. Access the disk containing the SSPGCS LOADLIB. This should be the SSP RUN disk.
2. Access the disk containing the NCP LOADLIB.
3. Add the SSPGCS and NCP load libraries to the GLOBAL LOADLIB statement in the VMVTAM EXEC.

GLOBAL LOADLIB VTAMUSER VTAM VSCS SSPGCS NCP
4. Create a file for the NCP load library using the command:

FILEDEF NCPLoad DISK NCP LOADLIB *

NCPLoad should be the same name as the load library name on the NCP BUILD statement.
5. Use the VARY command and the ATTACH command to attach the channel.
6. Use the VARY ACT command with LOAD=YES specified to load the controller.

MVS To load NCP into a communication controller from VTAM:

1. Include information on loading NCP in your start procedure:

:
//NCPLoad DD DSN=SYS1.NCPLoad,DISP=SHR
:
:
2. Use the VARY ACT command with LOAD=YES specified to load the communication controller.

VSE To load an NCP into a communication controller from VTAM:

1. Include information on loading an NCP in your VTAM start procedure:

//LIBDEF PHASE,SEARCH=(SYS1.NCPLoad)
2. Use the VARY ACT command with LOAD=YES specified to load the communication controller.

NCP Load Process

NCP LOADLIB is the load library containing the NCP load modules and resource resolution tables (RRTs).

When the NCP is loaded, the NCP definition statements, which are stored in VTAMLST, are read into storage on the host to define the resources connected through the communication controller. The NCP resource resolution table (RRT) is loaded into the host, and the NCP load module is loaded into storage on the communication controller. Figure 96 on page 363 illustrates the NCP load process.

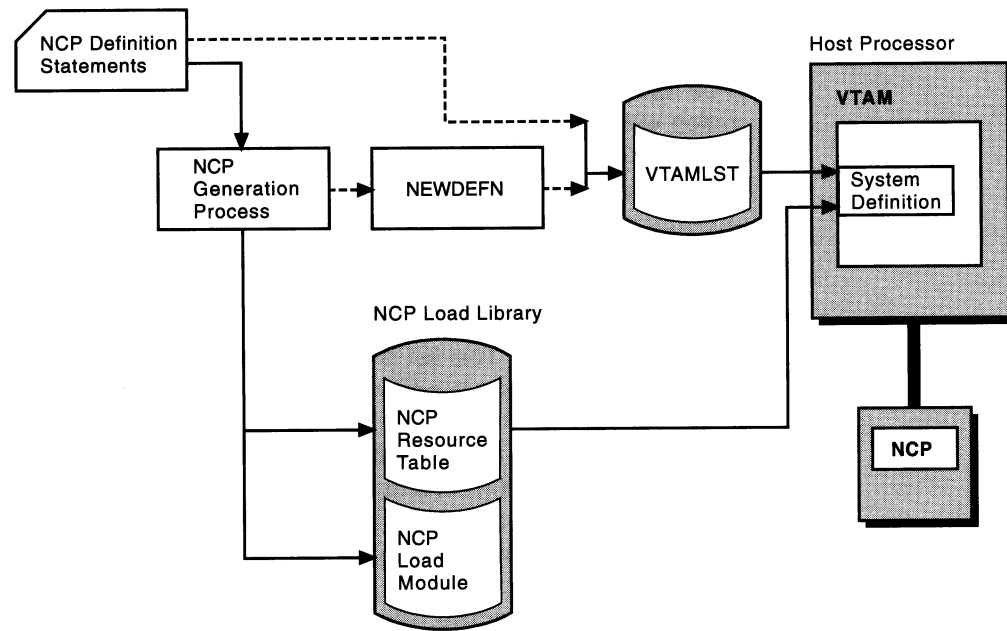


Figure 96. NCP Generation

The NCP definition statements are loaded directly into VTAMLST. However, if you generate a NEWDEFN data set, take the contents of the NEWDEFN data set and manually transfer them into VTAMLST. For more information about NCP loading and the NEWDEFN data set, see the *NCP, SSP, and EP Generation and Loading Guide*.

Identifying the Communication Controller

The communication controller into which the NCP is loaded is determined by the adjacent link stations specified for automatic activation with the NCP. These link stations in adjacent nodes locate the communication controller in the network.

If no link station is available, the load fails. In this case, specify `RENAME=adjacent link station name` on the command line.

Within its domain, VTAM ensures that no more than one load or dump operation is attempted at one time for a given NCP through any adjacent link stations that might be available.

If you are activating an NCP in a communication controller that already contains a different NCP, you should first deactivate the current NCP and any active link stations in adjacent subarea nodes that are connected to that NCP. You can then enter the command to activate the new NCP, which causes the communication controller to be reloaded as described on page 365 (if `LOAD=U` is specified or assumed). If adjacent link stations are not deactivated, the activation of the new NCP might fail because:

- There are no link stations available. (VTAM does not automatically break the existing connections with the old NCP.)
- The names of the old and new NCPs are the same, but their subarea addresses are different.
- The subarea addresses of the old and new NCPs are the same, but their names are different.

Selecting a Load Station

When loading an NCP, VTAM needs to know the name of the adjacent link station to be used. The load station can be either a channel link station or an SDLC link station within an NCP. VTAM can select a load station, or you can specify a load station. You specify the load station on the LOADSTA operand of the PCCU definition statement for the NCP or on the VARY ACT command, or during NCP error recovery if VTAM asks you whether to reload the NCP. VTAM selects the load station whenever there is no load station name currently in effect.

VTAM maintains a list of link stations over which a remote NCP can be contacted, loaded, or dumped. This list is maintained whether the NCP is active or inactive. This list of link stations is organized in a last-in-first-out (LIFO) order.

When activating the NCP, VTAM updates this list with any new link stations that are found in the NCP definitions or the VARY ACT command. VTAM searches this list for a suitable link to contact or load and then selects a link station by activating any link station that is not already active.

VTAM uses the first active link station it finds in the list. If it does not find an active link station by the time it reaches the end of the list, VTAM selects the last link station in the list and waits for it to complete activation before proceeding with the NCP activation.

If you want to override a load station name specified in an NCP definition, use the LOADSTA operand of the VARY ACT command to specify the name of another link station. You can also use this override capability to specify that VTAM should select the load station, despite the NCP definition. Do this by specifying a null value for the load station (that is, specifying the LOADSTA operand without a value). If you choose not to override the NCP definition value, VTAM uses that definition value to determine the link station to use.

When selecting a load station, the following restrictions apply:

- The load station must be specified for automatic activation with the NCP, or must be already active when it is needed (with VTAM thus knowing of its connection to the NCP). The former method is recommended.
- The link station must not be connected to a communication controller other than the one you want to load. Otherwise, you might load the wrong communication controller.
- The reconnection of cross-subarea links from one communication controller to another might require the changing of load station specifications for those NCPs intended to run in the communication controllers.
- Avoid the WARM start option and the WARM operand of the VARY ACT command when activating the affected NCPs if you change the load station specifications.
- Improperly specified LOADSTA values for two or more NCPs can result in a deadlock during their activation, with the activation of each NCP's load station waiting for the adjacent NCP's activation, which in turn can be waiting for the availability of a load station.
- You must define any link to be used as a load station link to both the NCP and the communication controller as being capable of initial program load (IPL):

- For the NCP, specify IPL=YES on the LINE definition statement in the NCP definition.
- For the communication controller, indicate the ability of a link to accept an IPL in the configuration file (CDF) residing in the communication controller's diskette storage.

Loading from the Host or from External Storage

For IBM 3720 and 3745 communication controllers, the VARY ACT command enables you to control whether the NCP is loaded from the host or from the 3720 or 3745 Communication Controller external disk storage.

The first NCP load can be from the host or a diskette. For more information on loading an NCP from a diskette, see the *NCP, SSP, and EP Generation and Loading Guide*. For communication controllers other than 3720s or 3745s, the NCP can be loaded only from the host.

Reloading an NCP

If VTAM is allowed to decide whether to reload the communication controller, one of the following occurs:

- If the communication controller contains no program, VTAM loads the NCP into the communication controller.
- If VTAM decides that a load is not necessary, VTAM issues a message to inform you, and it performs no disk-related functions.
- If the communication controller already contains the NCP being activated, you might be asked whether the existing NCP should be refreshed with a new copy. If you specified AUTOSYN=NO on the PCCU definition statement in the NCP definition, VTAM does not automatically re-synchronize itself with the existing NCP, but rather asks you whether re-synchronization or reloading should occur. Otherwise, VTAM re-synchronizes itself with the NCP without operator intervention and without reloading the communication controller.
- If the communication controller contains an NCP other than the one being activated, you might be asked whether the existing NCP should be replaced. If you specified VFYLM=YES on the PCCU statement in the NCP definition, you are asked to confirm that the communication controller should be reloaded. Otherwise, VTAM proceeds with the reload without operator intervention.
- If the communication controller contains a program other than an NCP (for example, an EP program in a channel-attached communication controller), the NCP activation fails. During activation VTAM attempts channel contact, but when the identification exchange fails, VTAM assumes that the communication controller might contain an EP program. VTAM then attempts to re-synchronize with the assumed NCP. Activation of the channel link station fails when the resynchronization fails. This results in failure of the NCP activation itself. Another VARY ACT command specifying LOAD=YES can be entered to force loading of the NCP.

Scheduled Automatic Reloading of a Communication Controller

To automatically IPL your IBM 3745 Communication Controllers at a scheduled date and time:

1. Set the date and time associated with the specific load module.
2. Maintenance operator subsystem (MOSS) sends an alert before starting the IPL (default is 60 minutes). The alert is not displayed by VTAM. The alert is forwarded to the NetView program and can be viewed using the NetView hardware monitor. You can have the NetView program send a message to the NetView operator when an alert is received by modifying the NetView program's automation table. For information on modifying this table, see *NetView Automation Implementation*. For information on the format of the ALERT, see *3745 Guide to Timed IPL and Rename Load Module*.
3. MOSS IPLs the load module from the communication controller disk at the specified time.

For example, if you want to have your communication controller IPLed at 1600 on 24 May 1993, then you would use the command:

```
MODIFY NET,LOAD,ID=ncpa,ACTION=SETTIME,NOTIFY=30,
IPLTIME=(05/24/93,16:00),LOADMOD=ncp1a
```

This command causes MOSS to load the load module *ncp1a* into NCP *ncpa* at 16:00 on 24 May 1993. This command also indicates that MOSS will send out an alert 30 minutes before the scheduled IPL (NOTIFY operand).

Functions of Scheduled Automatic Reload: You can also do the following when implementing this function:

Modify the format of the date

You can use the DATEFORM start option to modify the format you use to set the date. The default is MM/DD/YY. For the syntax of this start option, see "DATEFORM" in the *VTAM Resource Definition Reference*.

Specify the alert interval

You can change the length of the interval between the time MOSS sends an alert to the time that MOSS IPLs the communication controller. Use the MODIFY LOAD command with the NOTIFY operand. The default is 60 minutes.

Display a scheduled IPL

You can display the information about a scheduled IPL using the DISPLAY DISK command. For the syntax of this command, see "DISPLAY DISK Command" in *VTAM Operation*.

Cancel a scheduled IPL

You can cancel a scheduled IPL using the MODIFY LOAD command with ACTION=SETTIME,IPLTIME=CANCEL specified.

Example of a Scheduled Automatic Reload: Following is an example of a scheduled automatic reload of an NCP:

1. Load and activate the NCP using the options PUNAME=X, NEWNAME=A, and LOADMOD=A.
2. Change the original NCP generation or write a new NCP definition using the options NEWNAME=B and PUNAME=Y.

3. Load the changed or new NCP generation, store it on the hard disk, and set a time for it to IPL.
4. The NCP is reloaded at the specified time. A soft INOP is generated, and VTAM tries to bring up a session with the NCP. Because the PU name has changed, the session state will be pending ACTPU (PAPU2).
5. Even if you reloaded NCP with the same load module name, any new resources defined to NCP in the new NCP generation (which is now loaded in the NCP) will not be known to VTAM. Therefore, any attempts to work with these newly defined resources fails. For VTAM to be aware of the changes, use the VARY INACT command with the FORCE option to bring down the session between VTAM and NCP and then activate the session using the LOAD=NO option.

Note: If your NCP is in pending ACTPU state and you have an active virtual route between VTAM and the NCP through another VTAM, then the ACTPU is sent over this virtual route (rather than directly to the NCP over the original link). NCP responds with an ACTPU response, but the load module name returned is not the one VTAM is expecting. You will receive message IST605I, and the session between NCP and VTAM is deactivated. In this case, reactivate the session with the new load module name and the LOAD=NO option.

Considerations for Scheduled Automatic Reloading

- You cannot schedule two load modules for the same communication controller unit to be loaded within 5 minutes of each other.
- A 3745 Communication Controller may have two central controller units. You cannot load one central controller unit using a scheduled automatic reload while the other central controller unit is waiting to be loaded.
- If MOSS goes down after you have scheduled an automatic reload, one of the following occurs:
 - If MOSS returns online after the time for the scheduled reload, the scheduled reload is cancelled. No alert is sent.
 - If MOSS returns online before the time for the scheduled reload and the time to send the alert has passed, MOSS performs the reload as scheduled and an alert is sent if possible.
 - If MOSS returns online before the time to send the alert, MOSS sends the alert and performs the reload as scheduled.
- If you schedule an automatic reload with a load module that contains errors, MOSS attempts to load the load module again before failing the load.
- If you have more than one VTAM that owns a remote 3745 communication controller, consider the following:
 - It is possible for a person at each VTAM to set different scheduled automatic reloads for the same load module. If this happens, the scheduled automatic reload set by one VTAM can be written over by the other VTAM. MOSS performs the reload at the last time set for the load module.
 - If MOSS receives a command to purge a load module that has a scheduled automatic reload, MOSS cancels the scheduled reload and purges the load module. No message is sent to the VTAM that scheduled the reload.

- If MOSS receives a command to replace a load module that has a scheduled automatic reload, MOSS replaces the load module and cancels the scheduled reload. No message is sent to the VTAM that scheduled the reload.
- AUTODMP and AUTOIPL operands on the PCCU definition statement are always set to YES after a scheduled automatic reload. For more information about these operands, see “AUTODMP” and “AUTOIPL” in the *VTAM Resource Definition Reference*.
- If the time you specify for the IPL time is the same as the current system time, MOSS does not IPL.
- If you set the IPL time for more than 15 days in advance, you might encounter minor MOSS clock deviations. However, if this is not a problem, then you can set the IPL time up to 90 days in advance.

Renaming NCP Load Modules

You can change the name MOSS associates with the load module on the communication controller disk to another name. This is useful if you want to load from a disk or keep load module changes transparent to your operators.

Use the MODIFY LOAD command with the NEWNAME operand and ACTION=RENAME specified.

To associate the NCP name with different load module names without using the renaming step:

1. Generate the NCP source programs using different load module names.
2. Activate the NCP and specify the load module by using the LOADMOD operand.
3. Use the name specified in the ID operand, not the load module name, to identify the NCP.

Sample Load Module Name Change: Following is a sample scenario of renaming an NCP load module. In this scenario, one load module is being replaced by another. However, the final load module name will be the same as the original load module name.

1. Activate an NCP with load module NCPA and start the NCPA using the VARY ACT command:

```
V NET,ACT,ID=NCPA,LOADMOD=NCPA,LOAD=YES,SAVEMOD=YES
```
2. The NCP is activated and a load module NCPA is saved on the MOSS disk because you specified SAVEMOD. Now an NCPA member is in VTAMLST, an NCPA member is in NCPLOAD, and you have a session between VTAM and NCPA.
3. Copy NCPA to a file named NCPA1, add your changes, and generate the file on another system. On the other system, the VTAMLST member is called NCPA1. Because the NEWNAME parameter in the generation was not changed, but other keyword values in the generation were changed, the generation name of the NCPLOAD library member will be NCPA. Use an external MVS job to rename this member to NCPA1.
4. Move NCPA1 (the VTAMLST member) and NCPA1 (the NCPLOAD library member) to the original MVS system. Now you have two load modules:

- NCPA on the VTAMLST library and NCPA on the NCPLOAD library.
- NCPA1 on the VTAMLST library and NCPA1 on the NCPLOAD library.

The NCP name has not changed and the NEWNAME parameter in the generation has not changed.

5. Use the MODIFY LOAD command to add NCPA1:

```
F NET,LOAD,ID=NCPA,ACTION=ADD,LOADMOD=NCPA1
```

A non-disruptive ADD is performed.

6. Use the MODIFY LOAD command to rename the current load module on the MOSS disk:

```
F NET,LOAD,ID=NCPA,ACTION=RENAME,LOADMOD=NCPA,NEWNAME=NCPA2
```

7. Use the MODIFY LOAD command to rename the load module NCPA1 on the MOSS disk to NCPA:

```
F NET,LOAD,ID=NCPA,LOADMOD=NCPA1,NEWNAME=NCPA,ACTION=RENAME
```

8. Use the VARY ACT command to deactivate the session with the NCP:

```
V NET,INACT,ID=NCPA
```

9. Rename the member name NCPA on the VTAMLST and the NCPLOAD library to NCPA2 using an external MVS job. Then rename NCPA1 (the VTAMLST member) and NCPA1 (the NCPLOAD library member) to NCPA. You now have the following:

- NCPA2 member in VTAMLST (MVS job only, not generation)
- NCPA2 member in NCPLOAD (MVS job only, not generation)
- NCPA member in VTAMLST
- NCPA member in the NCPLOAD.

10. Use the VARY ACT command to activate the NCP and to load the NCP from the MOSS disk.

```
V NET,ACT,ID=NCPA,LOADMOD=NCPA,LOADFROM=EXT,LOAD=YES
```

Note: Remember, the NEWNAME in this new generation was never changed, so to VTAM the NEWNAME parameter is NCPA and is not different from the name of the load module NCPA (which is also the name of the NCP). Now you have:

- NCPA2 member in VTAMLST
- NCPA2 member in NCPLOAD
- NCPA member in VTAMLST
- NCPA member in the NCPLOAD.

Considerations for Changing Load Module Names: If you only rename a load module on the MOSS disk and do not change the VTAMLST and NCPLOAD library members, the following problems might occur:

- The load module does not activate if the name is changed only on the MOSS disk.

If you code the following, the activation of the load module will not occur:

```
F NET,LOAD,ID=NCPA,LOADMOD=NCPA,NEWNAME=NCPA2,ACTION=RENAME
```

```
V NET,ACT,ID=NCPA,LOADMOD=NCPA2,LOADFROM=EXT,LOAD=YES
```

VTAMLST or NCPLOAD library members have not been changed.

Note: VTAM searches the VTAMLST file for load module NCPA2. If VTAM cannot find it, VTAM rejects the command. VTAM does not activate NCPA, even if the VTAMLST and NCPLOAD library member names have changed to NCPA2, because the NEWNAME parameter in the VTAMLST will not have a matching NCP generation on the NCPLOAD library. Remember, you changed only the external member names; you did not change the NEWNAME parameter and generate the load module.

- The renamed load module cannot be replaced if it is renamed only on the MOSS disk. The following procedure will not work:

```
F NET,LOAD,ID=NCPA,LOADMOD=NCPA,NEWNAME=NCPA2,ACTION=RENAME
F NET,LOAD,ID=NCPA,ACTION=REPLACE,LOADMOD=NCPA2.
```

VTAMLST and NCPLOAD library members were never changed.

Note: VTAM processes the MODIFY LOAD command and tries to load NCPA2. VTAM looks for the NCPA2. Because VTAM does not find the NCPA2, VTAM fails the load. If the NCPA (the VTAMLST member) and NCPA (the NCPLOAD library member) are changed externally to NCPA2, the replace will occur.

Functions for Changing Load Module Names: You can do the following even if a load module is renamed on the MOSS disk. The assumption is that a load module has already been renamed. In the following cases, a load module NCPA has been renamed to NCPA2:

- Purge a renamed load module on the MOSS disk.

```
F NET,LOAD,ID=NCPA,LOADMOD=NCPA2,ACTION=PURGE
```

VTAM does not perform any checking for the VTAMLST library member NCPA2.

- Set an IPL time so that the load module can IPL at a certain desired time.

```
F NET,LOAD,ID=NCPA,LOADMOD=NCPA2,ACTION=S,IPLTIME=(4/28,11:30)
```

VTAM does not check its VTAMLST members and the NCP load library members. MOSS IPLs NCP at the required time.

- Rename a previously renamed load module.

```
F NET,LOAD,ID=NCPA,ACTION=RENAME,LOADMOD=NCPA#,NEWNAME=NCPA
```

VTAM does not check for the load module, and the load module is renamed.

Loading or Purging a Load Module on a Communication Controller Disk

You can use the MODIFY LOAD command to explicitly load a load module from the host into the 3720 or 3745 Communication Controller external disk storage while an NCP is already active in the communication controller.

At activation time, you can use the LOADMOD operand of the VARY ACT command to specify the load module to be loaded into the communication controller. After activation, the MODIFY LOAD command allows you to add, replace, or purge load modules on the 3720 or 3745 Communication Controller disk.

You can add a load module to the communication controller external storage by using the ADD operand of the MODIFY LOAD command. If the load module to be

added exists on the communication controller disk or if the maximum number of load modules allowed is already on the disk, the command fails.

You can replace a load module already on the communication controller disk by using the REPLACE operand of the MODIFY LOAD command. VTAM then replaces the load module specified on the communication controller disk. If the load module is not present on the disk, an ADD operation is attempted.

The REPLACE operand affects the automatic activation of a load module if an automatic re-IPL is required. If the load module being replaced is the load module tagged for automatic re-IPL, then after the REPLACE operation starts, the load module is no longer tagged to be automatically activated when an automatic re-IPL is initiated. If an automatic re-IPL occurs while a REPLACE operation is running, the communication controller is not activated with any load module.

The CANCEL option of the MODIFY LOAD command can be used to stop an ADD or REPLACE operation that is in progress. If the CANCEL operand is used, the partially stored load module is purged from the disk.

Activating an NCP

Although there is no resource hierarchy among major nodes, there are some considerations for NCP activation:

- There must be at least one defined route to the NCP; this requires that an appropriate path definition set be activated before the NCP is activated. NCP activation remains pending until a route to the NCP becomes physically available.
- **VM** Releases of NCP prior to Version 3 cannot be activated if activation requires that the NCP be loaded.

If the member name of the NCP source program differs from the name that you want to be associated with the NCP after activation, use the LOADMOD operand on the VARY ACT command to identify the member name in both VTAMLST and in the specified LOADLIB. The member name must be the same as the load module name.

Automatic Activation of Adjacent Link Stations

When activating an NCP, VTAM can be provided with a list of link stations in subarea nodes adjacent to the NCP that are to be automatically activated along with the NCP. Automatically activated link stations and already active link stations adjacent to the NCP connect the NCP to the rest of the network. In addition to providing access to the NCP being activated, activating the link stations provides link station ownership to the VTAM SSCP. The NCP containing the link stations supplies VTAM with notification of link station failures. Activation of the link stations enables you to load or dump the NCP through the link stations.

A VARY ACT command for an NCP major node can result in the automatic activation of cross-subarea links and link stations in adjacent subarea nodes (other NCPs or VTAM itself). For example, if you enter a command to activate an NCP, VTAM automatically activates the link stations in adjacent nodes that are known to be needed for the connection to the NCP being activated. VTAM knows this from the channel device name or RNAME specification (or both) of either the NCP definition statement or the VARY ACT command. However, the full route must be available for automatic activation to be completely successful.

Specifying Adjacent Link Stations: You can specify the link stations that are to be automatically activated using the CUADDR and RNAME operands on the PCCU definition statement for the NCP. (This method is recommended to simplify operating procedures and reduce the chance for operator error.) If you do not specify these link stations, or if you need to override the defined link station names, you can use the U or RNAME operands (or both) on the VARY ACT command when activating the NCP.

When the NCP is activated, the specified link stations and their associated links are automatically activated if they are not already active. If any of these link stations are not yet known to VTAM (because the NCP to which they are subordinate has not been activated), VTAM remembers them until they become known (through the activation of their major node). Then VTAM completes the activation. VTAM informs you of the link stations that are awaiting the activation of their NCP.

When activating a link in a channel-attachment major node, the operator needs to supply the channel device name of the link unless it is supplied in the LINE definition statement. The operator can also specify the channel device name to override the value in the LINE definition. To supply the channel device name, use the U operand of the VARY ACT command when activating the link.

If VTAM needs the operator's assistance to create a connection for a link in a channel-attachment major node, VTAM displays a message indicating that a connection request is denied because the link is not physically attached or is unknown to the operating system. In either case, the link must be available before the channel-attachment major node can be activated.

Example of Automatic Activation of Adjacent Link Stations: Figure 97 shows an example of this process. Assume that LSA is automatically activated when you activate NCP1. If you want to activate NCP2 through NCP1, first either manually activate LSB and LINKA or have them automatically activated (coded using RNAME=LSB).

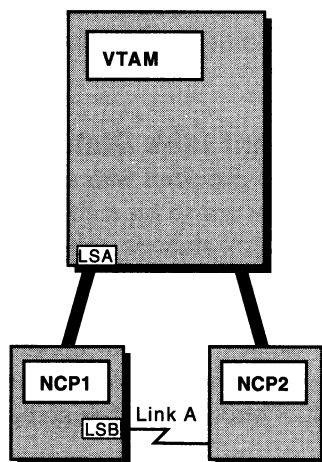


Figure 97. Activating NCPs and Link Stations: Through Another NCP

Going the other direction, if you want to activate NCP2, LSC must be active or automatically activated. If you want to activate NCP1 through NCP2, you must manually or automatically activate LSD and LINKA.

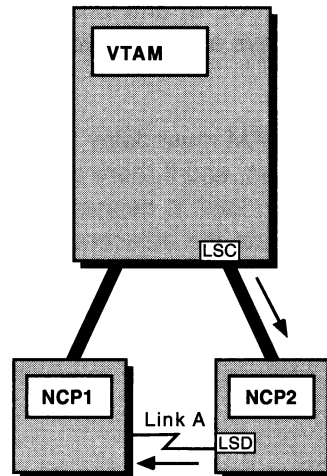


Figure 98. Activating NCPs and Link Stations: Through Host Link Station

If NCP1 was inactive when you tried to activate NCP2, the activation of LINKA and LSB would be delayed until NCP1 was activated. VTAM would communicate with NCP2 through its channel connection (LSC).

Following is an example of automatic activation of link stations.

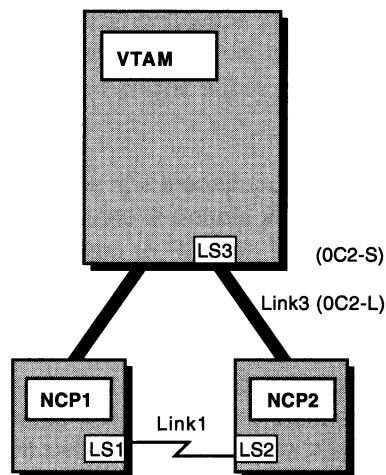


Figure 99. Automatic Activation of Link Stations

For Figure 99, the following is coded for NCP2:

```
CUADDR=0C2,
RNAME=LS1,
```

1. NCP1 is active. LINK1, LS1, and NCP are not.
2. You activate NCP2.
3. VTAM automatically activates LS3 (0C2-S) and LINK3 (0C2-L), enabling communication over LINK3. VTAM also automatically activates LS1 and LINK1, enabling communication through NCP1 over LINK1.

Following is an example of delayed automatic activation:

1. In Figure 99, NCP1, LINK1, LINK3, and NCP2 are inactive.
2. You activate NCP2.

3. VTAM automatically activates LS3 (OC2-S) and LINK3 (OC2-L), enabling communication over LINK3. VTAM delays automatic activation of LS1 and LINK1 until you activate NCP1.

Delay of Automatic Activation: If VTAM must delay the activation of all the link stations specified for automatic activation, and if there are no other active connections to the NCP, the NCP activation itself is delayed to await the availability of an active connection. You should activate the adjacent NCPs containing the link stations to allow the original NCP activation to complete. Similarly, if none of the routes to the NCP are physically available, VTAM displays a message to you and delays the activation of the NCP until at least one route becomes available. However, if a request is entered to activate an NCP to which no route has been defined, the activation fails.

Any suspended activation of an NCP or link station can be terminated by deactivating that NCP or link station.

Updating Adjacent Link Stations: If the physical links to a communication controller running NCP are reconnected to a different communication controller running another NCP, the link stations specified for automatic activation with the NCPs might need to be updated to reflect the new connections. That is, the CUADDR and RNAME values on the PCCU statement in the NCP definitions might need to be updated. Also, the WARM start option and the WARM operand of the VARY ACT command should be avoided when activating NCPs if the link station specifications are changed. Failure to update the adjacent link station specifications might result in the activation or loading of the wrong communication controller when an NCP is activated.

When VTAM automatically activates an adjacent link station during activation of an NCP, VTAM expects to find that the link station is connected to a communication controller that either requires loading of an NCP or contains the NCP being activated. If, through an operator or system programmer error, the list of link stations to be automatically activated includes one or more link stations that are connected to the wrong communication controller, VTAM might not detect the error and could assume that this wrong communication controller is to be reloaded with the NCP being activated. Therefore, you should ensure that link stations are correctly chosen for automatic activation with an NCP. Ensure that the link stations are in fact connected to the communication controller that does contain that NCP.

Reactivating a Failed NCP

If an NCP fails or is manually reloaded immediately before or during the activation of an NCP by VTAM, the activation might wait in a pending ACTPU state (PAPU1 or PAPU2). The adjacent link station must wait through its failure time-out period (specified on the ACTIVTO operand of the NCP GROUP definition statement) before it recognizes that its adjacent NCP has failed and that the link station is now inoperative. This timer value defaults to its maximum value of 7 minutes in the NCP generation process; so it can be up to 7 minutes before VTAM becomes aware that an SDLC-link-attached NCP has failed. VTAM bases its activation processing for the NCP during this period on the assumption that the NCP and its apparently active adjacent link stations are functioning normally.

You can recover by deactivating and reactivating the failed NCP. If it is certain that the NCP has failed and should be reloaded, you should specify LOAD=YES on the VARY ACT command for the NCP. If you do not specify LOAD=YES, any adjacent

link stations in surrounding NCPs should also be deactivated prior to the reactivation of the NCP. Then you can avoid the possibility of the timeout described previously. These link stations can then be reactivated automatically as part of the reactivation of the NCP, or they can be reactivated directly after the NCP has been successfully reloaded.

Activating Links

An SDLC, LAN, or X.25 link between two communication controllers can be activated directly, indirectly, or automatically.

How a Link Station Is Activated: Each cross-subarea link station is uniquely activated in the following ways:

- For channel link stations adjacent to an NCP, there are two methods of activation:
 - If a link station has been defined and named by VTAM, it can be activated only automatically. Any VARY ACT command specifying a channel link station is rejected. However, these channel link stations can be deactivated directly, indirectly (by halting VTAM), or automatically.
 - If the link station has been defined and named by the user (with a PU definition statement), it can be activated by the VARY ACT command.
- Channel link stations adjacent to another host processor can be activated only directly or indirectly, and can be deactivated with VARY commands. These channel link stations are named in the definition for the channel-attachment major node.
- SDLC, LAN, and X.25 link stations are activated directly or indirectly with the VARY ACT command.

When a cross-subarea link station becomes active, VTAM issues a message to the operator indicating the name and subarea number of the adjacent subarea node with which contact has been established.

Owning Link Stations: Activation of the link station makes the link station active to VTAM and makes VTAM's SSCP an owner of that link station. For cross-subarea SDLC, LAN, and X.25 link stations, more than one SSCP can share ownership of an NCP link station if the link station is activated by each SSCP. However, for cross-subarea SDLC, LAN, and X.25 link stations within a host processor node, only one SSCP can own and activate the link station.

An NCP link station can also be active without SSCP ownership if its link has been defined to use the SDLC monitor mode function in the NCP. As a result, a cross-subarea SDLC link station within an NCP can be capable of supporting session traffic without having been activated by VTAM.

Delayed Activation of Logical Lines: This function enables you to perform an orderly activation of a physical resource and its associated logical resources. Because there is a hierarchical relationship between the physical resources (line and physical unit) and the logical lines, you can use this function to implicitly activate a logical line when the physical resources are explicitly activated.

The function creates a hierarchy between a logical line in a group (within an NCP major node) that has a PHYSRSC operand coded and the physical resource indicated by the PHYSRSC operand. This includes:

- NCP token-ring interconnect (NTRI) physical resources (line and physical unit) and logical lines for INN link stations
- NCP Packet Switched Interface (NPSI) physical resources (line and physical unit) and logical lines
- Frame-relay physical resources (line and physical unit) and logical lines.

Lines within any line group for which the PHYSRSC operand is coded are logical lines. Lines within a line group for which PHYSRSC points to a valid physical unit within the same NCP major node are hierarchically subordinate to the physical unit. Not all logical lines are hierarchically subordinate to a physical resource. Because an NTRI switched peripheral line can be used serially for connections through different token-ring interface coupler (TICs), lines within a line group where the PHYSRSC, ECLTYPE, and DIAL=YES operands are coded are not subordinate to the physical unit. In this case, VTAM does not create the hierarchy between the physical resources and those logical lines. PHYSRSC must point to a physical unit to create the subordinate logical line relationship.

Activating logical lines hierarchically

A subordinate logical line can be activated only after its higher-level physical unit has been activated. The SCOPE operand on the VARY ACT command works with this hierarchy like it does with other hierarchical relationships in VTAM. For example, SCOPE=ALL on the higher-level physical unit causes the subordinate logical lines and their subordinate resources to be activated. SCOPE=U causes only the logical lines with ISTATUS=ACTIVE (and any subordinate resources with ISTATUS=ACTIVE) to be activated. SCOPE=COMP and SCOPE=ONLY also work with this hierarchy.

Deactivating logical lines hierarchically

When a logical line is subordinate to the physical resource, deactivation of the physical resource deactivates the logical lines.

Restrictions on Activation (VM, VSE)

You cannot activate an NCP under the following circumstances:

- If you have an SDLC link defined in a channel-attachment major node; such a link through a communication adapter can be used only to contact an NCP that is active in another domain.
- If the only RNAME given for the NCP is a link station associated with an SDLC line through a communication adapter.
- If the LOADSTA operand on the PCCU definition statement specifies a link station associated with an SDLC line through the communication adapter and if activation of the NCP requires that the NCP be loaded.
- If the NCP is connected by an SDLC link to a communication adapter, and both the NCP and the communication adapter are in the same domain, that communication adapter cannot activate that NCP.
- If you are working in an ES/9000 series environment, NCP activation fails if you issue an ACTPU to an NCP over the token-ring network.

Loading and Activating an NCP over a Switched Link

When a remote NCP is activated using the VARY ACT operator command, you can specify the switched link station name on the RNAME operand of the PCCU definition statement or on the RNAME operand of the VARY ACT operator command to have VTAM automatically perform the dial operation and establish the switched connection. The following examples show the two ways to specify the RNAME operand to load and activate the remote NCP2 in Figure 100 on page 378:

On the PCCU Definition Statement for NCP2

```
PCCU  SUBAREA=1,
:
      LOADSTA=NS1L1,
      RNAME=(NS1L1,SW1L1),
:
```

On the VARY ACT Operator Command

```
VARY ACT,ID=NCP2,LOAD=YES,RNAME=(NS1L1,SW1L1)
```

The load can be performed over either of the two possible link stations (NS1L1 or SW1L1) in Figure 100 on page 378. Because both link station names are specified on the RNAME operand, VTAM does the following:

- Activates the nonswitched link
- Performs the dial operation and establishes the switched connection.

The VTAM default load station preference is:

1. Active channel link
2. Active nonswitched link
3. Active dial-out switched link
4. Active dial-in switched link
5. Pending active nonswitched link
6. Pending active dial-out switched link
7. Pending active dial-in switched link.

Therefore, if the nonswitched link is operative, the load is performed using the non-switched link.

If you omit the switched load station (SW1L1) from the RNAME operand, the switched link connection is not established, and only the nonswitched connection is used. If a failure occurs on the nonswitched link, the operator can issue a subsequent VARY ACT operator command with the RNAME operand specifying the switched load station name (SW1L1).

If the switched link is not used for the initial activation and load process, you can activate it at any time using a VARY ACT operator command.

Note: When you are loading an NCP over switched lines and the NCP has channel-attached or leased line connections to another host, the NCP load is not complete until the host that initiated the command receives an indication that the command has completed.

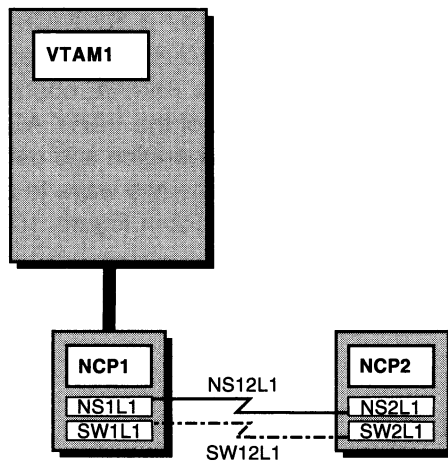


Figure 100. Loading and Activating NCP Using Switched Link

Enabling Communication Controller Channels for Activation

The channel-enable switches on a communication controller panel (or the MOSS console for the 3720 and 3745) determine whether VTAM can activate a channel link and link station during an NCP activation. For a communication controller that is both channel-attached and SDLC-link-attached, these switches also determine whether an NCP can be loaded over a channel or an SDLC link. If any channel adapters are enabled on the communication controller, the remote program load (RPL) feature of the communication controller is disabled, and the communication controller can be loaded only over one of these channels. If none of the channel adapters are enabled, the communication controller can be loaded only over SDLC links defined as being capable of IPL (previously described under "Selecting a Load Station" on page 364).

All channel adapters should be kept enabled during normal operation with channel-attached NCPs. This is because a channel-link station activation fails (with the channel appearing to be "not operational") if the channel adapter for the channel VTAM is attempting to use is disabled. VTAM does not recover the use of the channel (even if the channel adapter is later enabled) unless the NCP is deactivated and reactivated. If all channel adapters are enabled, the SDLC links are still available for normal use after the NCP is loaded or if the NCP has been previously loaded. The channel-enable switches should be turned off only when a communication controller must be loaded over an SDLC link (such as when the channel-attached host has failed along with the NCP).

Figure 101 on page 379 shows an example of an NCP that is both channel-attached and SDLC-link-attached to a VTAM host. For this example, assume that NCP1 is active and that NCP2 is inactive and requires loading of its NCP load module.

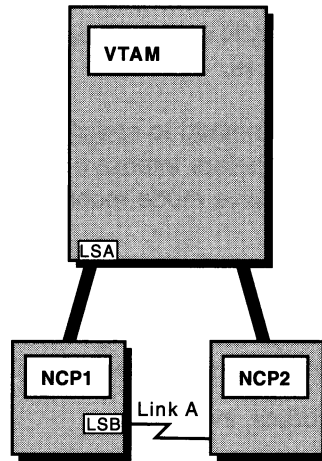


Figure 101. Loading over a Channel

You activate NCP2 with the following command (with NCP2's channel adapter enabled on the communication controller panel):

```
VARY NET,ACT,ID=NCP2,U=0C2,RNAME=LINKSTA1
```

VTAM then attempts to activate the adjacent link stations 0C2-S and LINKSTA1. Activation of LINKSTA1 in NCP1 is delayed because NCP1 receives no response to link-level commands sent on the SDLC link to NCP2's communication controller before and during the loading of NCP2 over the channel. Because the channel adapter is enabled, activation of 0C2-S proceeds, and VTAM begins the loading of NCP2 over the channel. After the load completes and LINKSTA2 is activated, contact is established with NCP1, and the activation of LINKSTA1 completes.

Monitoring an NCP

Monitoring your NCPs includes checking for hardware errors in devices attached to your NCPs, checking the contents of external disk storage, and testing links.

Sending Line Errors to VTAM

The NCP maintains statistical information regarding temporary line errors and other hardware error conditions for links and PUs. Normally the NCP sends this statistical information to VTAM only when a statistical counter overflows or a permanent error occurs. With intensive mode recording (IMR), error information is sent to VTAM as the errors occur. You can use the MODIFY IMR command to start IMR. With the MODIFY IMR command, you can request intensive mode recording for a communication link between an NCP and one of its peripheral physical units, or between two NCP major nodes.

The error information is sent to VTAM in a miscellaneous data recorder (MDR) record. MDR records are written to the system error log (LOGREC). If the NetView hardware monitor is in your system, VTAM can send copies of the MDR records to them.

To begin intensive mode recording on a link between an NCP and a peripheral physical unit, specify the name of the physical unit on the ID operand of the MODIFY IMR command. The SDLC link must be active.

To begin intensive mode recording on a link between two NCPs, specify the name of the link station in the NCP initiating IMR on the ID operand of the MODIFY IMR command. The SDLC link must be active.

You can also use the MODIFY IMR command to specify the number of intensive mode recording events to be recorded before intensive mode recording is automatically terminated. You can cancel intensive mode recording before this limit is reached.

Displaying NCP Storage

The DISPLAY DISK command displays the following information about the contents of IBM 3720 and 3745 Communication Controller external disk storage:

- Information about each load module, including:
 - The load module name
 - The date and time the load module was stored on the disk
 - Stability of the load module (stored, storing, or suspended)
 - An indication of whether the load module on the disk is the same as the load module currently active in the communication controller
- Information regarding dumps, including:
 - The dump name
 - The date and time the dump was written to the disk
- An indication of whether the load module is tagged for automatic IPL.

The DISPLAY NCPSTOR command displays:

- The storage contents of an NCP's communication controller
- The storage contents of an NCP's communication controller external disk storage (valid only for IBM 3720 Communication Controllers with the communication controller external disk storage feature or for the IBM 3745 Communication Controller)
- The state vector from the NCP's communication controller external disk storage (valid only for IBM 3720 Communication Controllers with the communication controller external disk storage feature or for the IBM 3745 Communication Controller).

This command displays up to 256 bytes, beginning at any address within the communication controller or communication controller external disk storage. The bytes are displayed on the VTAM operator's console.

Testing a Nonswitched SDLC Line

A link level 2 test is used to test a nonswitched SDLC communication line between:

- NCP and one of its peripheral PUs
- VTAM and one of its peripheral PUs
- Two NCPs
- Two VTAMs (The command must be initiated from the VTAM with the highest subarea number.)
- NCP and VTAM (The command must be issued from the NCP's domain.).

You can specify:

- Either a continuous link test or the number of test transmissions to be sent

- The cancellation of an ongoing link test
- Any user data to be included in the test transmissions.

The test data is returned from the remote station (NCP, host node, or peripheral PU) to the controlling NCP or to VTAM. The data received by the NCP or VTAM is compared with the data sent, and the following information is displayed by VTAM:

- Number of test messages sent
- Total number of test messages returned (including those in error)
- Number of test messages returned without error.

The SDLC link must be active, and the physical unit must be inactive. You must specify the name of the physical unit on the ID operand of the MODIFY LL2 command. You can run this test on a point-to-point line or on a multipoint line without affecting any other physical units on the line.

Before you request a link level 2 test between two NCPs, ensure that the link station at the originating end is inactive and the link station at the receiving end is active. In a multiple-domain network with shared host ownership of link stations, both link stations must be inactive to all SSCPs. The name of the link station in the NCP initiating the test must be specified on the ID operand of the MODIFY LL2 command.

Deactivating an NCP

Because there is no resource hierarchy among major nodes, any NCP major node can be deactivated at any time. The deactivation of an NCP does not cause VTAM to deactivate any other NCP. However, there are physical dependencies you must consider. When an NCP is deactivated, all paths using the NCP are also deactivated. Therefore, any NCPs that are link-attached to the NCP being deactivated are rendered inoperative. In addition, deactivation of an NCP or a cross-subarea link or link station can have significant and disruptive effects on other subarea nodes.

To avoid disrupting sessions, you need to be concerned about the status of cross-subarea links and link stations after an NCP deactivation.

Same-domain links are governed by the following rule: when you enter a VARY INACT command for an NCP, VTAM automatically deactivates any link or link station in an adjacent subarea node that was automatically activated when the NCP was activated. VTAM does not perform these automatic deactivations if you activate links and link stations directly or indirectly.

Deactivating Links

VTAM deactivates a cross-subarea link station adjacent to a host processor when the channel-attachment major node or the associated channel, SDLC, LAN, or X.25 link is deactivated (even if the associated link is currently carrying session traffic). Deactivating such a link station drops any sessions using a route going through that link station, possibly including the SSCP's session with another subarea node. Also, VTAM might be an intermediate routing node for cross-domain sessions that pass through the link station. Care should be taken not to deactivate a cross-subarea link station that could be supporting session traffic.

The NCP deactivates an SDLC, LAN, or X.25 cross-subarea link and its subordinate link station when the last of the link's SSCP owners deactivates the link.

When the last active link station in a transmission group is deactivated, any session traffic being carried by that transmission group is disrupted. Therefore, if the VTAM operator has not activated a cross-subarea link and link station, and VTAM is using that link's transmission group to carry session traffic, those sessions are disrupted when the link's owners all deactivate the link and link station and there are no other active link stations in the transmission group. On the other hand, if the VTAM operator activates the link and link station, VTAM shares the ownership of them, and the link and link station remain active, regardless of the deactivation requests of other domain operators.

If the VARY INACT command is used to deactivate a channel link owned only by VTAM, and that link's link station is the only active link station in a transmission group, deactivating that link destroys any sessions using a route containing that transmission group, possibly including the SSCP's session with another SSCP or with the NCP containing the link being deactivated. For this reason, care should be taken not to deactivate such a cross-subarea link.

Note: To prevent the automatic deactivation of cross-subarea NCP SDLC, LAN, or X.25 links in adjacent subarea nodes when an NCP is deactivated, you can directly or indirectly activate the links before the NCP is deactivated. The operator uses individual VARY ACT commands or VARY ACT with the SCOPE operand; the system programmer uses the initial status specification in the link definitions.

Deactivation of a cross-domain link also causes the deactivation of its subordinate link station.

Dumping an NCP

You can request either a dynamic dump or a static (destructive) dump of an NCP.

Dynamic dump

When you request a dynamic dump (MODIFY DUMP OPTION=DYNA), NCP processing continues while the NCP's contents are being dumped. A dynamic dump reflects the storage contents of the communication controller over a period of time.

Static dump

You request a static dump by entering the MODIFY DUMP command with OPTION=STATIC (the default value), or by replying "yes" when VTAM offers to dump a failed NCP. When you request a static dump, NCP processing stops and the current storage contents of the NCP are dumped. This dump effectively destroys a running NCP. The NCP must be reactivated (and reloaded) before communication with that NCP can resume.

The method for obtaining a static dump of an NCP depends upon whether the NCP is active or inactive.

Active NCP

If the NCP is active, specify the name of the NCP to be dumped. Select the dump station and dump file to use as described under "Selecting a Dump Station for a Static Dump of an NCP" on page 384 and "Selecting a Dump File for an NCP Dump" on page 385. As a result of the dump, the NCP is deactivated.

Inactive NCP

If the NCP is inactive, specify the name of an SDLC link station within an NCP adjacent to the communication controller containing the NCP to be dumped.

The link station specified on the command is the dump station. This link station can be either active or inactive, but the associated SDLC link must be active so that it can be used to carry the dump traffic. As a result of the dump, the link station is deactivated, if it is not already inactive. Because there is no dump file associated with a link station, specify the logical unit name (SYSxxx) on the DUMPDS operand on the MODIFY DUMP command.

VM To dump an NCP, you must do the following:

1. Access the disk containing the SSPGCS LOADLIB. This should be the SSP RUN disk.
2. Link and access a read/write disk as x.
3. Create a file definition for the NCP raw dump:

```
FILEDEF dumpds DISK filename filetype x
```

dumpds must be the same as the name specified on the DUMPDS operand of the PCCU definition statement.

4. Enter the MODIFY DUMP command.

Use this same procedure for MOSS and CSP dumps (except that *dumpds* must match the name specified on the MDUMPDS or CDUMPDS operands of the PCCU definition statement).

Although this method can also dump an active link-attached NCP, it is primarily used for dumping an inactive link-attached NCP. If an active NCP is dumped by specifying ID=*adjacent link station* on the MODIFY DUMP command, the adjacent link station, not the NCP, is deactivated when the dump is completed. To deactivate the NCP, issue the VARY INACT command.

You can also obtain a static dump by using the NCP independent dump utility if the NCP is in a channel-attached communication controller. For more information on the NCP independent dump utility, see the *NCP, SSP, and EP Diagnosis Guide*.

When you request a static dump by specifying ACTION=STORE, OPTION=STATIC on the MODIFY DUMP command, the NCP terminates. The contents of the NCP are dumped to the controller's hard disk only if the automatic dump/load was previously set (DUMpload=YES,SAVEMODE=YES,LOADFROM=HOST or DUMpload=YES,LOADFROM=EXT) and the dump slot on the MOSS controller's hard disk is empty. The load module stored on the controller's hard disk is reloaded, but you need to reactivate the NCP before you can resume communication with that NCP. To transfer an NCP dump stored on a controller's hard disk to the host, use the ACTION=TRANSFER operand on the MODIFY DUMP command.

Use the DISPLAY NCPSTOR command to display the contents of the NCP external disk dump file. To print the contents of the NCP dump file, use the NCP independent dump utility described in the *NCP, SSP, and EP Diagnosis Guide*.

MVS If an expiration date exists for the dump file and it has not yet been reached, do not respond to message IEC507D by entering REPLY xx,M. Enter

REPLY *xx,U* to override the expiration date and allow write access to the file. Entering REPLY *xx,M* causes the dump to be abnormally terminated by the operating system.

Selecting a Dump Station for a Static Dump of an NCP

When you take a static dump of an NCP, VTAM requires the name of the adjacent link station through which the dump operation is to be carried out. If you specify an SDLC link station name on the ID operand of the MODIFY DUMP command, VTAM uses that link station as the dump station. If you specify an NCP name, a choice of dump stations is possible. The following paragraphs describe the possible ways in which a dump station can be selected when the MODIFY DUMP command specifies an active NCP.

The dump station can be either a channel link station or an SDLC link station within an NCP. VTAM can select a dump station, or you can specify a dump station. Specify the dump station (DUMPSTA) in the PCCU definition statement, on the VARY ACT command, on the MODIFY DUMP command, or during NCP error recovery if VTAM asks you whether the NCP should be dumped. VTAM selects the dump station whenever there is no dump station name currently in effect from one of the above sources. VTAM selects a channel link station if one is available. If no channel link station is available, VTAM uses any available SDLC link station.

If you want to override a dump station name specified in the PCCU definition statement, you can use the DUMPSTA operand of the VARY ACT command to specify the name of another link station. This override capability can also be used to specify that VTAM should select the dump station. This is done by specifying a null value for the dump station (that is, specifying DUMPSTA without a value). If you choose not to override the PCCU definition statement value, VTAM uses that definition value to determine the link station to be used. The DUMPSTA operand of the MODIFY DUMP command can likewise be used to override a dump station name specified in the PCCU definition statement or on the VARY ACT command.

A link station chosen as a dump station must also be specified for automatic activation with the NCP, or must be already active when it is needed (with VTAM thus knowing of its connection to the NCP). The former method is recommended. If a specified dump station is not available for a dump operation resulting from a MODIFY DUMP command, the command fails. If a specified dump station is not available for a dump operation resulting from error recovery procedures for an NCP, VTAM either prompts you for a link station name or selects a link station from among those that are available.

As with the link stations specified for automatic activation with an NCP, the reconnection of cross-subarea links from one communication controller to another might require the changing of dump station specifications for those NCPs intended to run in the communication controllers.

Switched, X.25, and token-ring link stations are valid dump stations when you specify OPTION=STATIC,ACTION=STORE on the MODIFY DUMP command.

MVS,VSE If you change the dump station specifications, avoid the WARM start option and the WARM operand of the VARY ACT command when you activate the affected NCPs.

Selecting a Dump File for an NCP Dump

When you take a dynamic or static dump of an NCP, VTAM requires the name of the file in the host that is to receive the dump. You can specify the dump file in the NCP definition with the DUMPDS operand (on the PCCU statement), or you can specify it on the MODIFY DUMP command with the DUMPDS operand. The value you specify on the DUMPDS operand should be a logical unit name (SYSxxx). Use the symbolic device name that is coded on the EXTENT or ASSIGN statement for the DLBL statement that contains the file name for the dump data set. The command overrides the specification in the PCCU definition statement, or specifies a file if one was not specified.

Selecting the Disposition of an NCP Dump

Dynamic or static NCP dumps can be stored on the host or, for IBM 3720 or 3745 Communication Controllers, on the communication controller external disk storage. You can specify where the dump is to be stored (in the event of NCP module error) by using the ACTION operand on the MODIFY DUMP command. The command overrides the DUMPDS operand of the PCCU definition statement.

Other Communication Controller Dumps

In addition to an NCP dump, you can obtain two other dumps for an IBM 372x or 3745 Communication Controller:

- A maintenance operator subsystem (MOSS) dump
- A communication scanner processor (CSP) dump.

Unlike the NCP dump, these dumps are not requested by you. If a controller's MOSS fails, the MOSS is automatically dumped to the controller's MOSS disk (for the 3720 and 3745 Communication Controllers) or diskette (for the 3725 Communication Controller). Likewise, if the controller's CSP fails, the CSP is automatically dumped to the controller's hard disk. When either a MOSS or CSP dump is taken, a notification is sent to VTAM, alerting you of the failure.

You issue the MODIFY DUMP command to move the contents of the dump to a file in the host processor, where it can be formatted and printed. In addition, the MODIFY DUMP command lets you purge the dump from the controller's hard disk without transferring it to the host.

Use the MODIFY DUMP command to:

- Transfer a MOSS dump file, specify TYPE=MOSS and ACTION=TRANSFER or COMP.
- Transfer a CSP dump file, specify TYPE=CSP and ACTION=TRANSFER or COMP.
- Purge a MOSS or CSP dump file without transferring it to the host, specify TYPE=MOSS or CSP and ACTION=PURGE.

If you have the NetView program in your system, VTAM is notified that the MOSS or CSP dump transfer to the host has failed. The command facility then notifies you of the failure and gives you the option of requesting the transfer of the dump file to a file in the host processor. Either you or the command facility can issue the MODIFY DUMP command.

To print the contents of the MOSS or CSP dump file, use the NCP independent dump utility described in the *NCP, SSP, and EP Diagnosis Guide*.

Selecting a Dump File for a MOSS or CSP Dump

When transferring a MOSS or CSP dump file, VTAM requires the name of the file in the host that receives the dump.

For a MOSS dump, the dump file can be specified in the PCCU definition statement with the MDUMPDS operand of the PCCU statement or with the DUMPDS operand on the MODIFY DUMP command. You can use the DUMPDS operand to override the specification in the PCCU definition statement or to specify a file if one was not specified. If you do not use the DUMPDS operand on the MODIFY DUMP command, and the PCCU definition statement does not include a MDUMPDS specification, VTAM places the MOSS dump in the file specified for an NCP dump. That is, the MOSS dump is transferred to the file specified by the DUMPDS operand of the PCCU definition statement.

For a CSP dump, the dump file can be specified in the PCCU definition statement with the CDUMPDS operand of the PCCU statement or with the DUMPDS operand on the MODIFY DUMP command. You can use the DUMPDS operand to override the specification in the PCCU definition statement or to specify a file if one was not specified. If you do not use the DUMPDS operand on the MODIFY DUMP command, and the NCP definition does not include a CDUMPDS specification, VTAM places the CSP dump in the file specified for an NCP dump. That is, the CSP dump is transferred to the file specified by the DUMPDS operand in the PCCU definition statement.

To obtain a MOSS or CSP dump, follow the instructions in “Dumping an NCP” on page 382.

If available, the bit error rate (BER) log, component descriptor file (CDF), token-ring interface coupler (TIC) dump, and communication adapter dump are transferred by the MODIFY DUMP command from the hard disk along with the MOSS or CSP dump. If the CSP or MOSS dump is not present, the BER log, CDF, TIC dump, and communication adapter dump are still transferred from the hard disk, but the command fails because the dump you specified was not transferred.

Part 3. Subarea Networks and SNA Network Interconnection (SNI)

Chapter 7. Implementing a Subarea Network	389
Implementing a Subarea Network	389
Starting VTAM	391
Defining the Location of Cross-Domain Resource Managers by Coding	
Adjacent SSCP Tables	391
Defining the Location of Cross-Domain Resource Managers Dynamically	392
Specifying Time-Out Values for Locating Cross-Domain Resources	392
Identifying VTAMs in Other Domains (CDRMs)	393
Identifying Resources in Other Domains	395
Dynamic Definition of Cross-Domain Resources	395
Static Definition of Cross-Domain Resources	397
Adjacent SSCPs	398
CDRM Owner Verification for Cross-Domain Resources	406
Changing Ownership of Cross-Domain Resources	406
Type 2.1 Casual Connection	407
Defining a Type 2.1 Casual Connection	408
Considerations for Using Casual Connection	413
Routing in a Subarea Network	414
How Session Traffic Is Assigned to a Specific Route	414
How to Plan Routes in Your Network	418
Subarea Addressing	422
Establishing and Controlling SNA Sessions	427
Operating VTAM in a Multiple-Domain Subarea Network	427
Activating Resources	427
Deactivating Resources	428
Operating with an NCP	428
Links and Link Stations	429
Discontiguous Domains	432
Backing Up Resource Owners	432
Sharing an NCP	434
Verifying a Multiple-Domain Subarea Network	435
 Chapter 8. Connecting Multiple Subarea Networks	 437
Implementing a Multiple-Network Environment	437
SNA Network Interconnection	437
Defining Your Network	437
SNI Configurations	440
Starting VTAM	441
Start Options for Gateway VTAMs	441
GWSSCP Start Option for Non-Gateway VTAMs	442
Configuration Lists for Gateway VTAMs	442
Application Programs	442
Defining the NetView Program	442
Connecting Networks	443
Defining a Gateway VTAM	445
Defining a Gateway NCP	446
Converting a Non-SNI NCP to a Gateway NCP	447
Defining Native Networks	447
Defining Non-Native Networks	450

Dynamic Network Identification	451
Defining Cross-Domain Resource Managers	455
Defining Cross-Domain Resources	458
Session Initiation Request	459
Name Assumption	460
Predefined Cross-Domain Resources without Network Specification	462
Predefined Cross-Domain Resources with Network Specification	463
Dynamic Cross-Domain Resources	465
Defining Adjacent SSCPs	467
Types of Adjacent SSCP Tables	467
Deciding Whether to Code Adjacent SSCP Tables	468
Sample of Adjacent SSCP Tables for a Multiple-Network Environment	468
Request Routing	471
Dynamically Defined CDRSCs and Adjacent SSCP Tables	472
Alias Name Translation and Adjacent SSCP Tables	473
Cross-Network Routing	473
Network Address Structures	473
Network Naming Conventions	474
Controlling Paths for Interconnected Networks	475
Handling Class-of-Service Tables	477
Address Translation	479
Resource Name Translation	481
Alias Selection Function of the Session Management Exit Routine	482
NetView Alias Name Translation Facility	483
Establishing and Controlling SNA Sessions	490
Non-Native Network Type 2.1 Connections	490
Automatic Logon	491
Session Management Exit Routine	492
Session Accounting	492
Operating VTAM	493
Activating Resources	493
Gateway NCP and VTAM interaction	493
Terminating Sessions	493
Using the NetView Program for Network Management	493
Verifying a Multiple-Network Environment	494

Chapter 7. Implementing a Subarea Network

This chapter describes how to implement a subarea network.

To avoid repetition, this chapter builds upon Chapter 5, “Getting Started with VTAM - Overview” on page 125 and Chapter 6, “Getting Started with VTAM - Detail” on page 161. Therefore, you should be familiar with Chapter 5 and Chapter 6 before reading this chapter.

Implementing a Subarea Network

A single-domain subarea network contains one system services control point (SSCP) that controls all the resources in the network. The SSCP resides in VTAM in the host processor. Defining your VTAM network involves identifying your domain and defining the resources in that domain to the SSCP.

A multiple-domain subarea network contains more than one system services control point (SSCP). Control of the resources in the network is divided among the SSCPs. Defining your VTAM network involves identifying your domains and defining the resources in those domains to the SSCPs.

Note: A VTAM host contains a host CP, even if you have defined your VTAM as a subarea node and not as an APPN node. This host CP is used for management services transport and is not used for session setup or routing.

Figure 102 on page 390 is the same as the configuration in Figure 33 on page 126, except that this is a multiple-domain network because it includes another host. Note that NCP12 is now NCP21. This is for naming convention purposes.

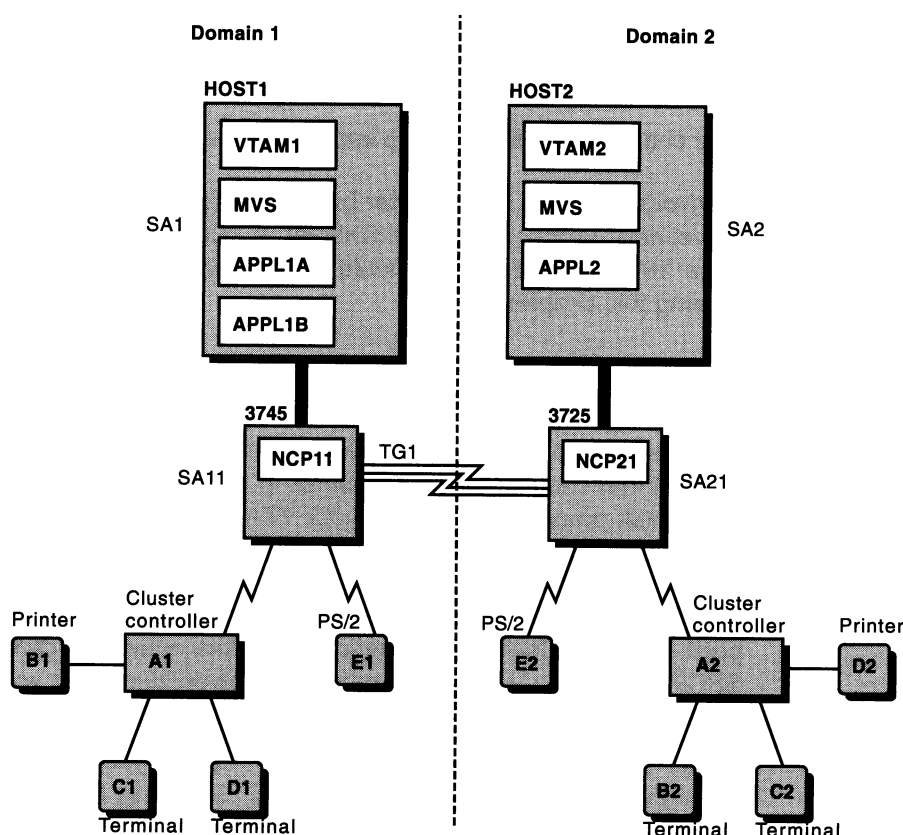


Figure 102. Multiple-Domain Network

Code everything that you would code to define the network in Figure 33 on page 126, except that you need to put the definitions for NCP21 and its peripheral nodes in HOST2 rather than in HOST1. You also need to code at least the following in HOST2:

- Start options describing HOST2 (NETID, HOSTSA, SSCPID, and SSCPNAME).
- One application program minor node defining APPL2.
- One NCP major node defining NCP21. This includes the NCP definitions. In NCP21's major node, also code another PATH definition statement to attach to subarea 2, and change the PATH definition statement for NCP 21 in NCP11's major node to DESTSA=21.
- VTAM PATH definition statements defining the explicit and virtual routes used for data flow throughout the network.
- One cross-domain resource manager major node containing two CDRM statements (one for HOST1 and one for HOST2).
- One cross-domain resource major node (or indicate the use of dynamic definition of cross-domain resources in the CDRM major node).
- One adjacent SSCP table (or indicate dynamic definition of adjacent SSCPs using start options [SSCPDYN and DYNASSCP]).

Also code at least the following in HOST1:

- The HOSTSA start option.

- A cross-domain resource manager major node containing two CDRM statements (one for HOST1 and one for HOST2).
- A cross-domain resource major node (or indicate use of dynamic definition of cross-domain resources in the CDRM major node).
- An adjacent SSCP table (or indicate dynamic definition of adjacent SSCPs using start options [SSCPDYN and DYNASSCP]).

Starting VTAM

In a multiple-domain subarea network, you can define the location of cross-domain resource managers and also let VTAM define cross-domain resource managers dynamically.

Defining the Location of Cross-Domain Resource Managers by Coding Adjacent SSCP Tables

In a multiple-domain network, session setup involves VTAM locating resources (session partners) in the network. Resources are located by routing a session initiation request to VTAMs in other domains. You can define a list of cross-domain resource managers (CDRMs) to which a session setup request can be routed. The list of CDRMs that you code for the search (referred to as adjacent SSCPs) can be supplemented dynamically by your VTAM domain. The search order, as well as the number of VTAMs to which the session request is routed, can affect session setup time. For more information, see “Adjacent SSCPs” on page 398.

VTAM provides two start options, SSCP DYN and SSCP ORD, that control how VTAM processes adjacent SSCP tables.

SSCP DYN start option

SSCP DYN determines whether VTAM should add entries dynamically to the adjacent SSCP table (that is, supplement the list of adjacent SSCPs that you have coded with other SSCPs that it discovers dynamically). The addition of these entries to the list can increase the time needed to search for a cross-domain resource.

If you code SSCP DYN=NO, you might have to define additional adjacent SSCP tables. When an application program passes a session (issues a CLSDST PASS macroinstruction) during session setup, VTAM does not build an adjacent SSCP table in the application program's network. When SSCP DYN=YES is specified in the application program's network, VTAM builds an entry in the table automatically.

SSCP ORD start option

SSCP ORD determines the order in which VTAM processes the entries in the table to route session requests to adjacent SSCPs. This start option indicates whether VTAM is to process the list in the order in which it is defined (DEFINED) or whether it is to route the session request to the last successful entry (PRIORITY). VTAM remembers successful and unsuccessful adjacent SSCP entries for routing session initiation requests. If you specify SSCP ORD=PRIORITY, VTAM always uses the last successful adjacent SSCP entry to route the session setup request.

One exception to order of search, regardless of the SSCP ORD value, is that VTAM always routes the session-initiation request directly to the

VTAM that owns the resource if it has an SSCP-SSCP session with that owner.

Defining the Location of Cross-Domain Resource Managers Dynamically

If you use the start option DYNASSCP=YES, you do not have to code an adjacent SSCP table. VTAM dynamically builds the list in the order in which the SSCP-SSCP sessions are activated. VTAM then uses this dynamically constructed list to route session initiation requests to the CDRMs, if an appropriate list is not defined.

Specifying Time-Out Values for Locating Cross-Domain Resources

The following start options also affect VTAM processing in a multiple-domain subarea network:

CDRSCTI start option

In a multiple-domain environment, you define resources that are controlled by a VTAM in another domain using a cross-domain resource (CDRSC) definition statement. However, if you do not code the CDRSC definition statement, you can specify that a CDRSC definition be dynamically created by using the CDRSC=OPT operand on the cross-domain resource manager (CDRM) definition statement that defines that VTAM. If your VTAM dynamically defines a cross-domain resource, that dynamic definition can be retained by VTAM for a user-specified period of time or deleted immediately after the last session for that cross-domain resource terminates. The cross-domain resource time-out value (CDRSCTI) start option determines the processing VTAM uses for dynamic cross-domain resources. The value of this start option specifies the amount of time that VTAM is to retain the control blocks for a dynamic cross-domain resource after all sessions with that resource are terminated. For more information, see “Dynamic Definition of Cross-Domain Resources” on page 395.

IOPURGE start option

During cross-domain session initiation, a CDINIT is sent to the adjacent node that is most likely to be the resource owner. If a positive response is received, session setup continues with that adjacent SSCP. If a negative response is received indicating the resource was not found, the next adjacent SSCP that potentially owns the resource is obtained from the adjacent SSCP table and a CDINIT is sent to it. This continues until the resource is either found or the adjacent SSCP table is exhausted.

A situation could arise where the adjacent SSCP that was queried has a critical failure and cannot respond. In this case, the node that is trying to initiate the session waits forever for a reply (IOPURGE=0, the default), thus blocking any future attempt to find the resource. Since the failing SSCP may not be the resource owner, the resource may be available and could be found if the search was allowed to continue through the adjacent SSCP table.

The IOPURGE start option can be used to specify how long VTAM will wait for a response before continuing a search. After a specified time interval, if a response has not been received, VTAM continues its search through the adjacent SSCP table until it either finds the resource or the table has been exhausted.

Identifying VTAMs in Other Domains (CDRMs)

A cross-domain resource manager (CDRM) is the part of an SSCP that supports cross-domain session setup and takedown. Before logical units in one domain can have cross-domain sessions with logical units in another domain, an SSCP-SSCP session must be established between the SSCPs of the two domains.

For an SSCP-SSCP session to exist, VTAM must know about all cross-domain resource managers with which it will communicate. You must define to VTAM its own cross-domain resource manager and all other cross-domain resource managers in the network. The cross-domain resource manager that represents the SSCP in your domain is called the host cross-domain resource manager. The cross-domain resource managers that represent the SSCPs in other domains are called external cross-domain resource managers.

Thus, to have a session between two SSCPs, define two cross-domain resource managers to each VTAM: one for the host and one for the external cross-domain resource manager. You file these definitions in a CDRM major node. Each cross-domain resource manager is a minor node.

One or more major nodes can be used to define cross domain resource managers. Each CDRM major node is defined with a VBUILD definition statement, and each minor node is defined with a CDRM definition statement.

Figure 103 shows two types of major and minor nodes in a multiple-domain network.

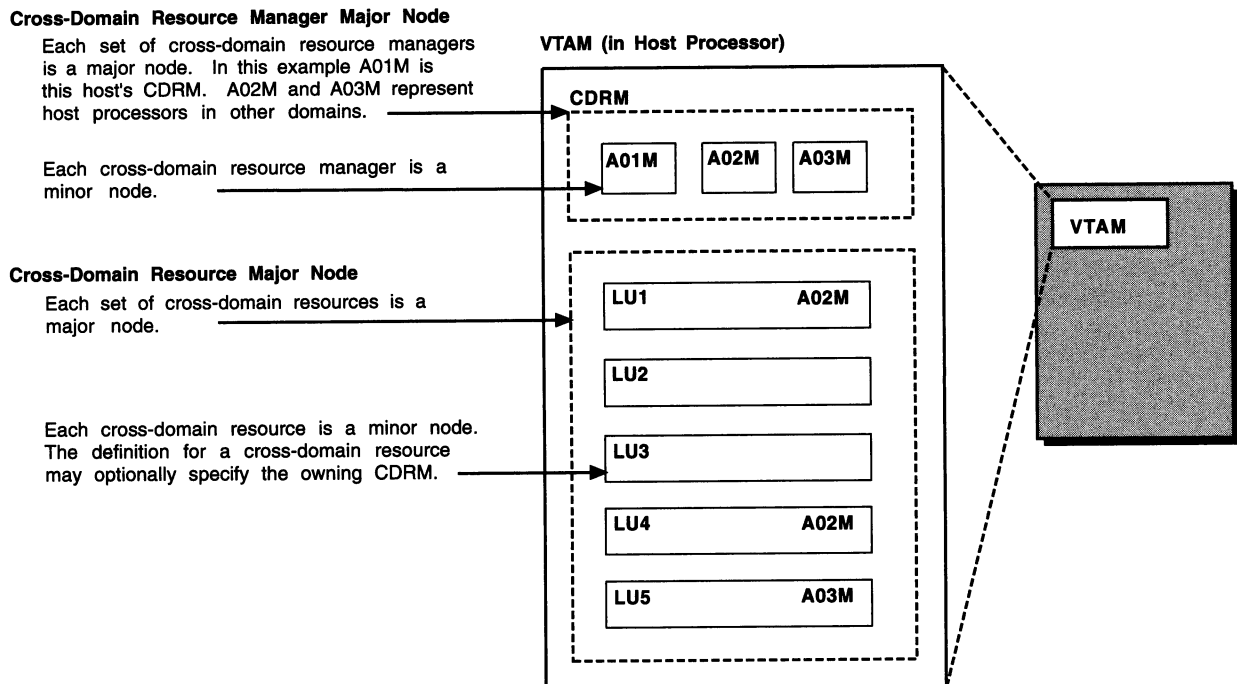


Figure 103. Major and Minor Nodes in Multiple-Domain Environment

Following is an example of major node CDRM for host A01M. The same definition set can be used at A02M and A03M. In this case, the definition set is at A01M. Thus, A01M is the host cross-domain resource manager and A02M and A03M are external cross-domain resource managers.

		VBUILD TYPE=CDRM	
A01M	CDRM	CDRDYN=YES, CDRSC=OPT, ELEMENT=1, ISTATUS=ACTIVE, SUBAREA=1, VPACING=63	AUTHORIZE DYNAMIC CDRSC ALLOW DYNAMIC CDRSCs CDRM ELEMENT NUMBER VTAM INITIAL STATUS NETWORK UNIQUE SUBAREA ADDRESS PACING BETWEEN CDRMS
*			
A02M	CDRM	CDRDYN=YES, CDRSC=OPT, ELEMENT=1, ISTATUS=ACTIVE, SUBAREA=2, VPACING=63	AUTHORIZE DYNAMIC CDRSC ALLOW DYNAMIC CDRSCs CDRM ELEMENT ADDRESS VTAM INITIAL STATUS NETWORK UNIQUE SUBAREA ADDRESS PACING BETWEEN CDRMS
*			
A03M	CDRM	CDRDYN=NO, CDRSC=REQ, ELEMENT=1, ISTATUS=ACTIVE, SUBAREA=3, VPACING=63	DYNAMIC CDRSC NOT AUTHORIZED REQUIRE PREDEFINED CDRSCs CDRM ELEMENT ADDRESS VTAM INITIAL STATUS NETWORK UNIQUE SUBAREA ADDRESS PACING BETWEEN CDRMS

Each CDRM minor node definition includes its network address and its initial status (active by default, or inactive). CDRM minor nodes are assigned to the subarea address of the appropriate host.

You can use the same set of CDRM definition statements for the entire network. Each VTAM uses the subarea specified on the CDRM definition statements to determine which definition statement defines its own host cross-domain resource manager. If the subarea on a CDRM definition statement matches the host subarea, VTAM uses the operands on this CDRM definition statement that apply to the host cross-domain resource managers and ignores operands that apply to external cross-domain resource managers. If the subarea on a CDRM definition statement does not match its host subarea, VTAM assumes that this CDRM definition statement applies to an external cross-domain resource manager. VTAM then uses only the operands that apply to external cross-domain resource managers.

When defining the host cross-domain resource manager, you can specify whether the host cross-domain resource manager is authorized to dynamically define cross-domain resources. If the host cross-domain resource manager and an external cross-domain resource manager are authorized in a VTAM domain, the other-domain SSCP can request a session setup for a logical unit for which no cross-domain resource (CDRSC) was defined. The receiving SSCP handles the request by dynamically creating a temporary CDRSC definition for the logical unit. Using dynamic definition of cross-domain resources can greatly reduce the number of definitions required in a multiple-domain network.

The default for the CDRDYN start option is YES, enabling dynamic definition of cross-domain resources. The CDRDYN operand can be coded on the host CDRM definition statement, but the CDRDYN start option overrides it. Also code CDRSC=OPT on external CDRM definition statements whose resources you want VTAM to dynamically define. The CDRSC operand is meaningful only on CDRM definition statements for external CDRMs.

The preceding example, which illustrates a CDRM major node, defines CDRMs that use different dynamic CDRSC options. The resulting capabilities of and restrictions on dynamic cross-domain resource definition are as follows:

- A01M can initiate session requests for any undefined cross-domain resources; however, it can route such requests only to A02M because that is the only other cross-domain resource manager with CDRSC=OPT. A01M can accept session requests from undefined cross-domain resources owned only by A02M because that is the only other cross-domain resource manager with CDRSC=OPT.
- A02M can initiate session requests for any undefined cross-domain resources; however, it can route such requests only to A01M. A02M can accept session requests from undefined cross-domain resources owned by A01M only.
- A03M cannot initiate or accept any session requests for undefined cross-domain resources.

Identifying Resources in Other Domains

This section describes the following:

- Dynamic definition of cross-domain resources
- Static definition of cross-domain resources
- How to locate cross-domain resources.

Resources controlled by a VTAM in another domain are called cross-domain resources (CDRSCs).

Note: CDRSC definition statements are also used to define independent logical units in the same domain and in non-native networks. See “Independent LUs” on page 140.

Dynamic Definition of Cross-Domain Resources

You do not have to define resources controlled by VTAMs in other domains. VTAM can dynamically create the definition statements to represent resources that reside in other domains. The operands on the CDRM definition statement for the host CDRM and the external CDRM determine whether VTAM can dynamically create CDRSC definition statements.

Notes:

1. You can also dynamically define independent logical units within the same domain using this function, and independent logical units in non-native networks. See “Dynamic Definition of Independent LUs” on page 221.
2. To define two or more cross-domain destination LUs that have the same name but are in different networks, use the NQNMODE=NQNAME start option. Otherwise, name conflicts cause session requests involving these LUs to fail.

To have resources in other domains dynamically defined to VTAM:

1. Code your host CDRM with CDRDYN=YES.
2. Code your external CDRM with CDRSC=OPT.

Now all unidentified resources residing under that external CDRM will be dynamically defined to your VTAM (host CDRM). Following is a sample coding for dynamic definition of resources in another domain:

```
          VBUILD TYPE=CDRM
MYHOST   CDRM  CDRDYN=YES,                      X
          SUBAREA=1
HOST2    CDRM  CDRSC=OPT,                        X
          SUBAREA=2
```

Dynamic definition can occur in the following two directions:

- When VTAM receives a session request from an undefined cross-domain resource, the cross-domain resource is the originating logical unit for this session. For example, a terminal in another domain attempts to log on to an application in your domain. The external cross-domain resource manager initiates the session, sending a session request to your VTAM. If the terminal is not defined to your VTAM as a cross-domain resource, VTAM does not accept the session request unless the host CDRM definition is coded with CDRDYN=YES and the external CDRM is coded in your VTAM with CDRSC=OPT.
- When VTAM sends a session request for an undefined cross-domain resource, the cross-domain resource is the destination logical unit for this session. For example, a local terminal attempts to log on to an application in another domain. If the application is not defined as a cross-domain resource, VTAM can still attempt to establish the session by sending a session request to adjacent SSCPs. VTAM sends the session request only if the host CDRM is coded with CDRDYN=YES and the adjacent CDRMs to which VTAM routes the session request are coded with CDRSC=OPT.

When VTAM creates dynamic CDRSCs for destination logical units, it must then dynamically locate the domain in which the resource resides. For this type of dynamic cross-domain resource, VTAM uses a search mechanism to route a cross-domain session request to other adjacent VTAMs. The mechanism is called an adjacent SSCP table and is discussed in "Static Definition of Cross-Domain Resources" on page 397 .

Dynamically defined cross-domain resources are collected in a CDRSC major node named ISTCDRDY. ISTCDRDY is activated automatically during VTAM initialization, and deactivated automatically during VTAM termination.

The dynamic cross-domain resource definition occurs when you have defined the host to allow dynamic definition of CDRSCs.

In general, you have the same control of dynamically defined cross-domain resources as predefined cross-domain resources. You can display and deactivate dynamically defined cross-domain resources using operator commands.

You also generally have the same control of the CDRSC major node ISTCDRDY as other CDRSC major nodes. The DISPLAY ID, MODIFY CDRM, VARY ACT, and VARY INACT commands can all be used for the ISTCDRDY major node.

While the host CDRM is active, the operator can deactivate ISTCDRDY with the VARY INACT command, in which case all dynamically defined CDRSCs are also deactivated and the dynamic CDRSC definition function is disabled. The dynamic

CDRSC definition function remains disabled until ISTCDRDY is activated again. You can activate ISTCDRDY again by issuing a VARY ACT command naming ISTCDRDY directly, or by activating the host CDRM (even if the host CDRM is already active). If the host CDRM is inactive, or if the host CDRM is not defined to allow the dynamic CDRSC definition function, activating ISTCDRDY does not provide the dynamic CDRSC definition function.

Dynamically defined CDRSCs are deactivated and deleted by VTAM on a periodic basis if they are not in use, based on the setting of the timer specified in the CDRSCTI start option. A dynamically defined CDRSC is also deleted if the host that owns that resource fails and the shadow resource for the CDRSC becomes the active definition. For more information on shadow resources, see “Shadow Resources” on page 407 .

Static Definition of Cross-Domain Resources

Cross-domain resources are logical units (application programs, peripheral nodes, and terminals) that are controlled by another VTAM domain. You define cross-domain resources in one or more cross-domain resource major nodes. Each cross-domain resource represents a minor node.

You define a cross-domain resource major node with one VBUILD definition statement for the major node and one CDRSC definition statement for each cross-domain resource in the major node. The same set of CDRSC definition statements can be used throughout the network.

The name on the CDRSC definition statement represents the name for the resource that is controlled by another VTAM. The CDRM operand, if used, specifies the name of the VTAM that controls that resource. If you do not specify the CDRM operand, the name of the other VTAM that controls the resource is not known. Therefore, you need to use an adjacent SSCP table, so that your VTAM can locate the CDRSC in another VTAM domain. An adjacent SSCP table is simply a list of other VTAMs with which your VTAM can have an SSCP-SSCP session. This list of VTAMs is used to search your network for the CDRSC. For more information about adjacent SSCP tables, see “Adjacent SSCPs” on page 398.

To statically define two or more CDRSCs that have the same name but are in different networks, use the NQNMODE=NQNAME start option to allow network-qualified names. For more information, see “NQNMODE” in the *VTAM Resource Definition Reference*.

In the following example of a CDRSC major node, some of the CDRSC definition statements do not specify a CDRM operand, so an adjacent SSCP table is required. However, some of the other cross-domain resource definition statements identify the CDRM owner.

```

*
* EXTERNAL VTAM APPLICATIONS – CDRSCS
*
          VBUILD  TYPE=CDRSC          CDRSC MAJOR NODE
*
* EXTERNAL VTAM APPLICATIONS – CDRSCS FOR VTAM SUBAREA NODE 03
*
NJE03    CDRSC                      JES/NJE
NETV03    CDRSC                      NETVIEW
CICS03    CDRSC                      CICS
NVAS03    CDRSC                      NETVIEW ACCESS SERVICES
*
* EXTERNAL VTAM APPLICATIONS – CDRSCS FOR VTAM SUBAREA NODE 70
*
S04VSCS  CDRSC  CDRM=A04M          VSCS
S04RSCS  CDRSC  CDRM=A04M          RSCS

```

Note: **MVS** Information on coding the CDRSC definition statement for TSO/VTAM is in “Multiple-Domain Network” on page 613.

A dependent logical unit and a CDRSC with the same network-qualified name can coexist. In a backup and recovery situation where one host is assuming ownership of a logical unit from another host in the same network, the dependent logical unit can be activated by the new host even though an application program within it currently has a cross-domain session with the logical unit. If the physical and logical units being recovered support ACTPU(ERP) and ACTLU(ERP) requests, the sessions with the logical units are not affected when the physical units and logical units are activated. The CDRSC definition automatically becomes a shadow resource, and the logical unit is now defined as a same-domain (APPL or LU) resource. If the current host wants to relinquish ownership of the logical unit, it releases or deactivates the logical unit and its associated physical unit. VTAM then makes the logical unit a shadow resource, and the CDRSC again becomes the active definition.

MVS When you define your network resources for XRF, consider the following:

- If you have predefined all CDRSCs in the network, you need to predefine a CDRSC having the USERVAR name and predefine a CDRSC for both the active and the backup application program.
- If you allow dynamic CDRSCs and trial and error rerouting, you do not need to do anything.

Regardless of whether you are using USERVARs with XRF, the names of your USERVARs must be unique. To ensure proper session setup with the desired partner, there can be no other resource within the network that has the same name as any of your USERVARs (except when the USERVAR is identical to its value).

Adjacent SSCPs

To supplement VTAM's session request routing service, you can code adjacent SSCP tables or have them dynamically defined. The adjacent SSCP table contains lists of SSCPs that can be in session with a host VTAM and can be used to search for the VTAM that controls a destination CDRSC.

When VTAM receives a session initiation or directory search request for a resource that is not located in that VTAM's domain, it attempts to locate the resource by

sending the request to its adjacent SSCPs. VTAM routes the request to the SSCP coded in the CDRM operand of the CDRSC definition statement. If one is not coded, the CDRSC is dynamically defined, or the session request with the SSCP in the CDRM operand fails, VTAM uses its adjacent SSCP table.

You can supplement the VTAM search with an adjacent SSCP table even though you code a CDRM operand. VTAM simply concatenates the proper adjacent SSCP list to the CDRM owner specification and uses the entries for the search. A session request for the CDRSC is routed to each SSCP in the list until the resource is located or the list is exhausted.

An adjacent SSCP table is particularly useful in networks where the SSCP ownership of resources changes frequently. VTAM can locate the owners dynamically rather than by having the programmer maintain the CDRSC definitions. Also, in networks where a large number of cross-domain resources are owned by only a small number of SSCPs, coding an adjacent SSCP table in each host can be easier than coding individual CDRSC definition statements.

If you code adjacent SSCP tables or have them dynamically defined, you do not have to code CDRSC definition statements for logical units in other domains, but VTAM's performance is slower because of the time it takes to send startup requests to SSCPs that do not own the logical unit. For information on how to improve the performance of adjacent SSCP tables, see "Improving Performance" on page 404.

Alternatively, a coded CDRSC can provide direct session setup, rather than waiting for VTAM to locate a logical unit using the adjacent SSCP tables. By placing CDRSC definitions at a central host (for instance, a CMC host) and placing single entry adjacent SSCP tables in other hosts that point to the central host, session setup requests are sent to the central host if nothing is known about a target resource. When the request reaches the central host, if there is a predefined real CDRSC coded with the resource name, the same NETID as the central host (the CDRSC definitions at the central host must have the same network identifier as the central host), and CDRM equal to the actual owning CDRM (note that the central host does not actually confirm that the coded CDRM is the owner), that information is returned with volatile USERVAR characteristics to the origin SSCP, which can rebuild its adjacent SSCP table with the owning SSCP as the first entry (similar to USERVAR processing). The session setup request can then be forwarded directly to the correct target.

You can also increase control over adjacent SSCP selection by creating adjacent SSCP lists for CDRSCs in an adjacent SSCP table. Then, if an adjacent SSCP list is identified for a CDRSC, session setup requests are sent to only the SSCPs in that list. If the owning SSCP is not found in that list, session establishment fails.

For information on using adjacent SSCP tables for routing requests to other networks, see "Defining Adjacent SSCPs" on page 467.

Adjacent SSCP Lists for CDRSCs

To define an adjacent SSCP list for a CDRSC, include the ADJLIST operand on the CDRSC definition statement. You can also code the ADJLIST operand on the GROUP definition statement, and the ADJLIST operand sifts down to any CDRSCs in the group for which ADJLIST is not coded. The ADJLIST operand specifies the name of a list of adjacent SSCPs, defined in an adjacent SSCP table using the ADJLIST definition statement followed by any number of ADJCDRM definition statements to identify the specific adjacent SSCPs in the list.

For example, following is a sample adjacent SSCP list that could be used by A02M to locate LU1 shown in Figure 104 on page 402.

```
A02ADJ      VBUILD TYPE=ADJSSCP
** ADJACENT SSCP LIST: THIS LIST IS USED WHEN THE ADJLIST OPERAND
**                                     FOR A CDRSC SPECIFIES LIST1
**
LIST1      ADJLIST
A01M      ADJCDRM
```

Following is a sample CDRSC definition statement for LU1.

```
                VBUILD TYPE=CDRSC
LU1          CDRSC ADJLIST=LIST1
```

When an adjacent SSCP list is identified for a CDRSC, session setup requests are sent to only the SSCPs in the list. If the owning SSCP is not found in the list, session establishment fails. This enables strict control over adjacent SSCP selection.

When using adjacent SSCP lists for CDRSCs, the SSCP selection function of the session management exit routine can be used to delete or reorder entries in a list. VTAM cannot dynamically add entries to an adjacent SSCP list.

To add, delete, or change an adjacent SSCP list for a CDRSC, use the MODIFY RESOURCE command. Adjacent SSCP lists for a CDRSC are displayed using the DISPLAY ADJSSCPS command. For details on these commands, see *VTAM Operation*.

Defining Adjacent SSCP Tables

You can define adjacent SSCP tables by using one, or both, of the following ways:

- Statically define the tables using VTAM definition statements.
- Specify the DYNASSCP start option as YES so that VTAM dynamically builds the tables.

One or more adjacent SSCP tables can be activated in each VTAM host.

Dynamic Definition of Adjacent SSCPs: VTAM can dynamically create an adjacent SSCP list. This table consists of all of the adjacent SSCPs with which VTAM has an SSCP-SSCP session. VTAM can then route session-establishment requests to all active adjacent SSCPs until the correct SSCP is found. If you code adjacent SSCP tables and use dynamically defined adjacent SSCP tables, VTAM uses the appropriate adjacent SSCP table without being affected.

To enable dynamically defined adjacent SSCP tables, code the DYNASSCP start option. If you specify YES and VTAM does not locate an appropriate adjacent

SSCP table, VTAM dynamically routes the session-establishment request to all active adjacent SSCPs until the correct SSCP is found. If you specify NO, VTAM does not perform this dynamic routing.

By using the dynamic adjacent SSCP table function in conjunction with the adjacent SSCP selection function of the session management exit routine, you can control the order of the dynamic adjacent SSCP table. See "Improving Performance" on page 404.

Static Definition of Adjacent SSCPs: Following is an example of an adjacent SSCP table. This table is coded in host A01M, whose configuration is illustrated in Figure 104 on page 402.

```
A01ADJ      VBUILD TYPE=ADJSSCP
** DEFAULT SSCP LIST: THIS LIST IS USED WHEN THE CDRM IS
**              UNKNOWN AND THE NETWORK IS UNKNOWN
**
A02M      ADJCDRM              DEFAULT SSCP#1
A03M      ADJCDRM              DEFAULT SSCP#2
**
**
**              NETWORK NETID=NETA
**
** ADJACENT SSCPS FOR A CDRM – USED IF CDRSC OWNER IS A02M
**              AND NETID IS NETA
**
A02M      CDRM
A02M      ADJCDRM              ROUTE ONLY TO THIS HOST
**
** ADJACENT SSCPS FOR A CDRM – USED IF CDRSC OWNER IS A03M
**              AND NETID IS NETA
**
A03M      CDRM
A03M      ADJCDRM              ROUTE ONLY TO THIS HOST
```

The *cdrmname* used on the adjacent CDRM definition statement in the adjacent SSCP table must be the same as:

- The name on the CDRM definition statement that defines the SSCP to VTAM.
- The SSCPNAME used in the VTAM or the name on the host CDRM definition statement in the external VTAM host.

In the SSCP of the originating logical unit, the CDRM name of the destination logical unit, if known, must be correct. If the CDRM name of the destination LU is not correct, the operator must issue a MODIFY CDRM command to change the CDRM name.

If SSCPORD=DEFINED and network specific tables have been coded, then session requests for DLU resources are not guaranteed to succeed unless at least one entry in the network specific table provides a connection to the DLU.

During adjacent SSCP routing for the DLU resource, SSCP entries are taken from either the network specific table or the default table. If the NETID of the DLU resource is known and a network specific table is coded for that NETID then the that table will be used. Otherwise entries from the default table will be used when the NETID is not known.

Network NETA

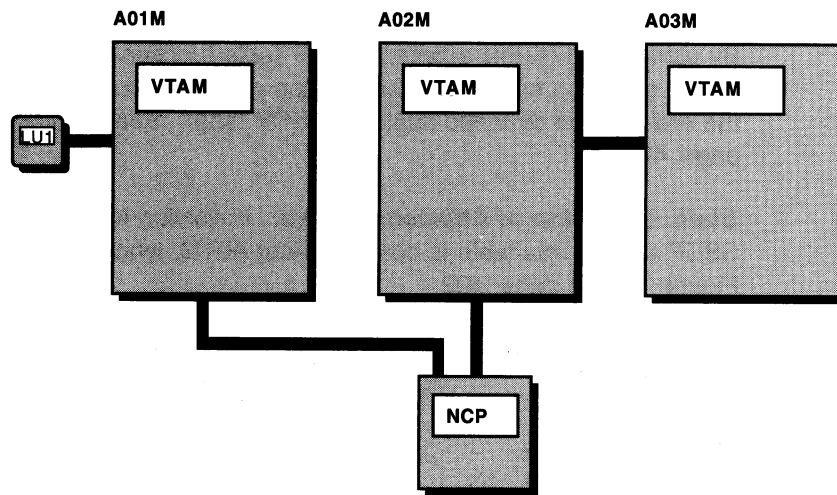


Figure 104. Example of Adjacent SSCP Table Connection

It is possible to code an entry in the default table that has a connection to the DLU and not code an equivalent entry in the DLU's network specific table. If the start option SSCPORD has the value of DEFINED then a session request for an unknown DLU would be successful because of the working entry in the default table. Subsequent session requests will now use the network specific table because the DLU netid is known. If the network specific table (matching the netid of the DLU) does not contain entries which provide a connection to the DLU, then the routing for these subsequent sessions will fail.

Table Search Order

You can individually activate multiple adjacent SSCP table definitions. One or more adjacent SSCP tables can be activated in each VTAM host. Each new set of definitions is added to the internal table used by VTAM. New definitions for destinations already in the internal table replace the current adjacent SSCP list for those destinations, but you cannot deactivate existing adjacent SSCP lists. If a new table replaces an existing one, VTAM issues a message informing the operator.

If the CDRM operand is coded on the CDRSC definition statement, it is used in an adjacent SSCP list. If the CDRM name is an entry in the list, it is moved to the top of the list. If the CDRM name is not an entry in the list, VTAM places the CDRM name at the top of the list. The list is then used to send session setup requests to SSCPs in the list until either the owning SSCP is found or the end of the list is reached. When the owning SSCP is found, VTAM remembers its name. None of these actions affect the actual adjacent SSCP table; they affect only the internal copy of the table that is maintained for each CDRSC.

The adjacent SSCP tables are searched in the following manner:

1. If the NETID and CDRM name are known, a search is done for a table coinciding with the known information.
2. If that table is not found or if only NETID is known, a search is done for a table defined with only the NETID given.

3. If an attempt to find that table fails, or if only the CDRM name is known, the indicated CDRM adjacent SSCP table for all networks is used.
4. If an attempt to find that table fails, the default adjacent SSCP table for all networks is used.
5. If an attempt to find that table fails and if dynamic adjacent SSCP tables are permitted, the dynamic table is used.

Routing with Adjacent SSCP Tables

If you have defined an SSCP list, VTAM sends the session setup request for an undefined destination logical unit to each SSCP in the list until either the owning SSCP is found or the end of the list is reached. If VTAM does not have an SSCP-SSCP session with an adjacent SSCP in the list, the SSCP is ignored. If a session setup request is routed to an SSCP that does not own the destination logical unit, the request is rejected.

For reliability during session setup, VTAM does the following:

- Associates the adjacent SSCP with a cross-domain resource whenever it receives a session initiation request from an SSCP that is not already in the table
- When looking for an SSCP with which to initiate a session, scans the adjacent SSCP table in priority order, giving preference to the following:
 - SSCPs for which the most recent session initiation attempt succeeded
 - SSCPs for which no session initiation attempts have been made.

VTAM bypasses any SSCP in the list with which it does not have an SSCP-SSCP session. If the owning SSCP is not found, the session request is rejected, and the dynamic CDRSC is deleted immediately. VTAM does not wait for the time value specified in the CDRSCTI start option.

Routing Rules: Session initiation and INQUIRE APPSTAT requests are routed from the originating LU's SSCP to the destination LU's SSCP using trial-and-error routing. There are several different means of determining the list of adjacent SSCPs to try when performing trial-and-error routing. However, after the list is determined, the processing is very similar for each of these requests. The rules for routing session requests are as follows:

- The request is not routed if its visit count (SSCP rerouting count) is 0.
- The request is never routed back to the SSCP from which it was received.
- The request is never routed back to its originating SSCP.
- The request is never routed in this network if it was received from an SSCP in this network.
- The request is never routed back to the network that it was received from through the same gateway NCP.

Routing and Dynamic Tables: If you specify the dynamic adjacent SSCP table function and you have not defined adjacent SSCP tables, VTAM dynamically routes session initiation requests to every SSCP with which it is currently in session. This feature eliminates the requirement to define adjacent SSCP tables. You can, however, continue to define adjacent SSCP tables. VTAM uses the user-defined table if one is defined.

Activating Adjacent SSCP Tables

Activate adjacent SSCP tables using the VARY ACT command. The ID operand of the VARY ACT command specifies either an adjacent SSCP table or a default SSCP list. Adjacent SSCP tables are not major nodes and cannot be deactivated.

Improving Performance

For performance reasons, you might want to change the way VTAM routes session requests. You can use either or both of the following to improve SSCP search performance:

- Use the adjacent SSCP selection function of the session management exit routine to shorten or reorder the list of adjacent SSCPs to which an LU-LU session request is directed. The list is composed of the adjacent SSCPs that are coded in the ADJSSCP table for a specific network or a default list and the list of adjacent SSCPs that was built during the previous LU-LU session setup requests. This exit routine can also be used in conjunction with the dynamic adjacent SSCP facility to avoid having to define any ADJSSCP table while maintaining control of session routing requests.
- Code the SSCPDYN and SSCPORD start options.

If the CDRM name is coded for the destination CDRSC, that entry is still tried first.

Note: When a resource name that is not valid is entered in a logon request, the request is not rejected until the initiate request has been sent to every SSCP in the default SSCP list and every SSCP has rejected it.

Adjacent SSCP Selection Function: VTAM constructs a dynamic adjacent SSCP table in the order in which the CDRM-CDRM sessions are established. Therefore, the search order for a resource might not be the same as that which can be achieved using a statically defined adjacent SSCP table. Extra session establishment time might be required if the SSCP list contains many entries or if it is arranged so that several SSCPs must be tried before the owning SSCP can be located. However, to control the order of search in this case, you can use the session management exit routine (adjacent SSCP selection function) in conjunction with a dynamic adjacent SSCP table.

In the session management exit routine, the adjacent SSCP selection function for LU-LU sessions allows you to modify the list of SSCPs in the search. You can shorten or reorder the list of SSCPs from which the next SSCP used in session setup is chosen during session request routing.

The list that the session management exit routine receives as input is either an adjacent SSCP table or a list of adjacent SSCPs that is built during the previous LU-LU session setup for the same resource. If no adjacent SSCPs exist, this function is not invoked. If you do not provide this function in your session management exit routine, VTAM uses the adjacent SSCP table as it exists.

“Adjacent SSCP Selection Function (Function Code X'06)” in *VTAM Customization* describes the adjacent SSCP selection function of the session management exit routine.

SSCPORD and SSCPDYN Routing Methods: Use the following table to help you decide how to set SSCPDYN and SSCPORD in your system.

SSCPDYN=YES SSCPORD=PRIORITY	VTAM adds entries dynamically to the adjacent SSCP table. When establishing sessions in releases prior to VTAM V3R2, VTAM gives priority to SSCPs for which the most recent session initiation attempt succeeded or for which no attempt has been made, even if a more direct path to the CDRM that owns the destination LU becomes available. When establishing sessions, VTAM V3R2 and later releases give priority to the CDRM that owns the destination LU (if known), then to SSCPs for which the most recent session attempt succeeded. This combination of options gives you the greatest flexibility for setting up routes across networks, and, if your adjacent SSCP table is large, it gives you the best performance during session setup.
SSCPDYN=YES SSCPORD=DEFINED	Entries are added dynamically to the adjacent SSCP table, but VTAM always searches the table from top to bottom when it tries to initiate a session. Because the dynamic entries are added at the bottom of the list, they are tried last during session initiation. This combination of options results in increased session setup time if the primary paths (those near the top of the list) are not available. But if a secondary path (one nearer the bottom of the table) is used, it does not get priority for subsequent sessions.
SSCPDYN=NO SSCPORD=PRIORITY	No entries are added to the adjacent SSCP table, except when the owner of a cross-domain resource is defined explicitly by the CDRM operand of the CDRSC definition statement. Therefore, VTAM can use only the paths you define in the table, and if one of these paths fails, VTAM does not attempt to use it again until all other paths also fail. This combination of options tends to limit the number of available cross-network routes. You might also need to define additional adjacent SSCP tables because VTAM does not build them automatically each time an application program issues a CLSDST PASS macroinstruction for a cross-network session.
SSCPDYN=NO SSCPORD=DEFINED	No entries are added to the adjacent SSCP table, except when the owner of a cross-domain resource is defined explicitly by the CDRM operand of the CDRSC definition statement. VTAM always searches the table from top to bottom when it tries to initiate a session. Again, VTAM can only use the paths you define in the table; but, because VTAM always searches the table from top to bottom, a path that fails can be used again as soon as it is restored. This combination of options results in increased session setup time if the primary paths (those near the top of the list) are not available. But if a secondary path (one nearer the bottom of the table) is used, it does not get priority for subsequent sessions. You might also need to define additional adjacent SSCP tables because VTAM does not build them automatically each time an application program issues a CLSDST PASS macroinstruction for a cross-network session. VTAM does not add new entries unless it is the owner of the resource. The owner is determined by the CDRM operand of the CDRSC definition statement, or by session initiation.

CDRM Owner Verification for Cross-Domain Resources

You can enable SSCPs to takeover cross-domain resources without requiring intermediate SSCPs to issue a MODIFY CDRM command. This enables VTAM to automatically change the CDRM ownership of a CDRSC. This can be useful in cross-network sessions where one of the session partners has been taken over by a different SSCP and an attempt is made to establish a new session.

Note: Owner verification of a cross-domain resource is optional.

This SSCP takeover of cross-domain resources is accomplished in the definition of the CDRSC with the VFYOWNER operand. VFYOWNER=YES can be specified only for a CDRSC that is coded with the NETID and CDRM operands. Specifying VFYOWNER=NO (which is the default) allows VTAM to automatically change the CDRM ownership of a CDRSC. VFYOWNER=YES in the CDRSC definition makes VTAM reject session setup requests that contain a conflicting owner. The default (VFYOWNER=NO) allows the takeover of a CDRSC without the need for intermediate SSCPs to issue a MODIFY CDRM command.

The MODIFY CDRM command does not need to be used for dynamically defined CDRSCs. If a session setup request is received for a dynamically defined CDRSC from a CDRM other than the current CDRM owner, VTAM automatically changes the CDRM ownership of the CDRSC. However, the operator might still want to use the MODIFY CDRM command if LU-LU session recovery is initiated by the logical unit that received the original session request.

For example, a terminal user logs on to an application program, and the CDRSC representing the terminal is defined dynamically in the domain of the application program. If the session between the application program and the terminal is disrupted, the application program might try to reinitiate the session instead of requiring the terminal user to log on again. In this case, the VTAM operator in the domain of the application program needs to use the MODIFY CDRM command to change the CDRM owner of the CDRSC representing the terminal, if the terminal is taken over by a backup SSCP other than the SSCP in the application program's domain.

Note that a VTAM domain that takes over a cross-domain resource does not need to use the MODIFY CDRM command because in this case, VTAM's shadow resource function applies. For more information on shadow resources, see "Shadow Resources" on page 407 .

Changing Ownership of Cross-Domain Resources

Each cross-domain resource (CDRSC) definition statement indicates its controlling SSCP (CDRM=*cdrmname*). The controlling SSCP is also called the owning CDRM of the cross-domain resource. The ownership of a cross-domain resource can be changed if the owning CDRM loses its SSCP-LU session with the logical unit, and another SSCP takes over the logical unit. For information on resource takeover, see "Steps for Resource Takeover" on page 432.

You can use the MODIFY CDRM command to inform VTAM of the takeover, naming the new owner of the cross-domain resource. This command can be used whether the cross-domain resource is active or inactive. If the cross-domain resource is already in session with a logical unit, the session is undisturbed and the owner is saved until the session terminates. Session startup requests continue to be routed to the old owner. Specifying TYPE=IMMED on the MODIFY CDRM

command can be used to force the ownership change immediately regardless of active sessions.

Shadow Resources

VTAM's shadow resource function enables VTAM to dynamically change what it defined previously as CDRSCs into same-domain resources during takeover.

A shadow resource is an alternate definition of a network resource. For example, an LU is defined to a backup host both as a same-domain resource and as a cross-domain resource. Under normal operation, the LU is a cross-domain resource, and the cross-domain definition is used. The same-domain resource definition, which is not used, is the shadow resource. Whichever definition is not being used is considered to be the shadow resource.

If the host that owns the LU fails, the backup host assumes ownership of the LU, the same-domain definition becomes the active definition, and the cross-domain definition becomes the shadow resource. Resource takeover can be accomplished without disrupting existing active sessions. When the original host becomes available, the backup host relinquishes ownership of the LU. The backup host then activates its cross-domain definition for the LU, and its same-domain definition once again is the shadow resource.

When the LU is owned by the backup host, other hosts that need to access the LU must be able to communicate with and route requests to the backup host.

Type 2.1 Casual Connection

Casual connection occurs when you define any of the following as type 2.1 peripheral nodes to each other, instead of using the usual VTAM subarea connection (with VTAM as type 5 and NCP as type 4):

- VTAM to VTAM
- VTAM to VTAM with NCP
- VTAM with NCP to VTAM with NCP.

Table 25 indicates the release levels of VTAM and NCP that can be defined to each other as type 2.1 peripheral nodes.

Table 25. Levels of VTAM and NCP Capable of Type 2.1 Casual Connection

	VTAM V3R3+	VTAM V3R3+ with NCP V4R3+ or NCP V5R2+	VTAM V3R2 with NCP V4R3+ or NCP V5R2+
VTAM V3R3+	Yes	Yes	Yes
VTAM V3R3+ with NCP V5R3	Yes	Yes	Yes
VTAM V3R2 with PTF and NCP V5R3	Yes	Yes	Yes

Note: The symbol + means "and later releases."

The NCP at the secondary end of a composite node must be NCP V5R3 or later.

An advantage of using a type 2.1 casual connection is that you eliminate the need to coordinate and define the following support for a subarea connection:

Type 2.1 Casual Connection

- Cross-domain resource manager (CDRM) connections, because the VTAM or VTAM and NCP are not defined using the standard type 5 and type 4 physical unit connections
- Logical units on the other side of the casual connection are defined as independent logical units. That means they can use multiple independent logical unit paths and dynamic independent logical unit definitions. See “Multiple Connections between a Type 2.1 Node and a Subarea Node” on page 224.
- Because the adjacent subarea node appears as a type 2.1 node, it can use dynamic switched definitions to define the switched type 2.1 node and dynamic independent logical unit definitions to define the logical units. For information on dynamic switched definitions, see “Defining Switched Resources Dynamically” on page 143.
- Adjacent SSCP tables because no CDRM definition statements are used and resources are always defined as residing at the adjacent type 2.1 node
- Subarea path definitions using explicit and virtual routes because the connection is defined only as a type 2.1 peripheral node attachment and not a subarea node (type 4 or 5) connection
- Operator procedures for establishing a switched subarea connection because VTAM performs the dial-out operation without operator intervention when an application program initiates a session.

Recommendation: Though casual connection offers some advantages over the normal subarea connection, it is not generally recommended that you use this technique to replace subarea network connections. The considerations associated with a casual connection restrict the connectivity of applications, dependent logical units, and NetView network management facilities.

Restrictions: Casual connection can only be used for PLU-initiate-only sessions and autologon sessions. There is no support for SSCP-SSCP type sessions, and you cannot queue sessions over a type 2.1 link.

Defining a Type 2.1 Casual Connection

To support a casual connection between VTAM and NCP subareas, you must ensure that the following definition requirements are used (similar to any other type 2.1 node):

- The physical unit definition for the NCP or directly attached VTAM must indicate exchange identification support (XID=YES).
- If you are predefining your independent logical units, you should use CDRSC definitions with ALSLIST entries describing the type 2.1 connection. Otherwise, you can dynamically define your independent logical units. See “Multiple Connections between a Type 2.1 Node and a Subarea Node” on page 224.
- For a switched connection, when the type 2.1 peripheral node is a VTAM, the CPNAME operand must be used. For VTAM to locate the physical and logical unit definitions, the value of the CPNAME operand on the switched line definition must be the same as the value of the SSCPNAME start option in the attaching VTAM type 2.1 peripheral node. The IDBLK and IDNUM operands are ignored.
- You need to know the maximum number of sessions associated with all of the type 2.1 nodes attached to the subarea (VTAM or NCP). For NCP type 2.1 connections, this number is defined during NCP generation and cannot dynam-

ically expand. However, for a direct VTAM connection, the session control blocks are held in the BSBUFF buffer pool that can expand and contract as necessary.

- For an NCP connection, you should also know the maximum number of sessions that any particular type 2.1 node can establish. If you do not specify this limit in NCP, a single independent LU in a type 2.1 node can exhaust all of the session control blocks generated in the NCP. There is no corresponding control for type 2.1 nodes directly attached to a VTAM. However, because this is a buffer pool, it can expand as needed to accommodate more sessions.
- When an independent LU in a type 2.1 node acts as the primary logical unit by initiating a session, NCP must assign another network address to the PLU. This address pool is generated in NCP. Because this is a pre-defined pool, you should know the maximum number of PLU sessions that all type 2.1 node independent LUs attached to that NCP can concurrently establish. For a direct VTAM connection, the addresses are simply assigned from the element pool (32768 elements) of the VTAM subarea.

In addition to the XID specification on the LINE definition statement, the MODE operand indicates that the VTAMA type 2.1 peripheral node is the primary and the VTAMB, the secondary SDLC station. VTAMA therefore performs the polling operation.

Figure 105 on page 412 illustrates other VTAM casual connections. HOSTC and NCPC are defined as a type 2.1 peripheral node to HOSTA, an ES/9000 series processor. HOSTA is also defined to HOSTC and NCPC as a type 2.1 peripheral node. The connection between the two nodes is a non-switched SDLC link. However, any of the following types of connections can be used:

- Nonswitched SDLC link
- Switched SDLC link
- X.25
- Token ring.

The other type 2.1 casual connection in Figure 105 on page 412 is between the two nodes (VTAMB with NCPB to VTAMC with NCPC). The channel connection between the VTAM and NCP in each case is a subarea node connection, a physical unit type 5 (VTAM) to a physical unit type 4 (NCP). The connection between the type 2.1 peripheral nodes is a switched SDLC link.

The application programs, APPLB and APPLC, are all defined as independent LUs in an adjacent type 2.1 node. APPLB appears to VTAMC as an independent LU in a type 2.1 node, which comprises VTAMB and NCPB. Conversely, APPLC appears as an independent logical unit to VTAMB and NCPB operating in a type 2.1 peripheral node (VTAMC and NCPC).

Following is an example of VTAM and NCP definition statements that are required for the type 2.1 casual connections shown in Figure 105 on page 412.

VTAMA Application Program

```

          VBUILD TYPE=APPL          APPLICATION PROGRAM
APPLA    APPL
:

```

Type 2.1 Casual Connection

VTAMA Nonswitched SDLC Connection

SDLCATT	VBUILD	TYPE=CA	
GROUP03	GROUP	LNCTL=SDLC, DIAL=NO	SDLC LINE CONTROL NONSWITCHED CONNECTION
LINE3	LINE	ADDRESS=033, MODE=PRI	LINE ADDRESS PRIMARY LINK STATION
PUC	PU	ADDR=A1, PUTYPE=2, XID=YES	CONTROLLER ADDRESS TYPE 2.1 PERIPHERAL NODE EXCHANGE ID PROCESSING

⋮

Independent Logical Units as Cross Domain Resources

CDRSCG1	VBUILD	TYPE=CDRSC	
APPLC	CDRSC	ALSLIST=PUC	INDEPENDENT LOGICAL UNIT
APPLB	CDRSC	ALSLIST=PUC	INDEPENDENT LOGICAL UNIT
APPLA	CDRSC	ALSLIST=PUA	INDEPENDENT LOGICAL UNIT

VTAMB Application Program

	VBUILD	TYPE=APPL	
APPLB	APPL		APPLICATION PROGRAM

⋮

VTAMB Switched SDLC Connection

SWNCPC	VBUILD	TYPE=SWNET	
PUC	PU	PUTYPE=2, ADDR=C1, CPNAME=VTAMC	TYPE 2.1 PERIPHERAL NODE CONTROLLER ADDRESS CONTROL POINT NAME IDENTIFIER
PATHC	PATH	DIALNO=1234567, GRPNM=NCPCSW	TELEPHONE NUMBER NCP SWITCHED GROUP NAME

⋮

NCPB Switched SDLC Connection

NCPB	PCCU		
⋮			
NCPCSW	GROUP	LNCTL=SDLC, DIAL=YES	SDLC LINE CONTROL SWITCHED CONNECTION
TONCPC	LINE PU		

⋮

VTAMC Application Program

	VBUILD	TYPE=APPL	
APPLC	APPL		APPLICATION PROGRAM

⋮

VTAMC Switched SDLC Connection

SWNCPB	VBUILD	TYPE=SWNET	
PUC	PU	PUTYPE=2,	TYPE 2.1 PERIPHERAL NODE
		ADDR=C1,	CONTROLLER ADDRESS
		CPNAME=VTAMB	CONTROL POINT NAME IDENTIFIER
PATHC	PATH	DIALNO=1234567,	TELEPHONE NUMBER
		GRPNM=NCPBSW	NCP SWITCHED GROUP NAME
:			

NCPC Nonswitched and Switched SDLC Connections

NCPC	PCCU		
:			
*			
** ES/9000 SERIES NONSWITCHED SDLC CONNECTION			
*			
	GROUP	LNCTL=SDLC,	SDLC LINE CONTROL
		DIAL=NO	NONSWITCHED CONNECTION
ES9000	LINE	ADDRESS=034	LINE ADDRESS
		MODE=SEC,	SECONDARY LINK STATION
		XID=YES	EXCHANGE ID PROCESSING
*			
PUA	PU	ADDR=C1,	LINE ADDRESS
		PUTYPE=2	TYPE 2.1 PERIPHERAL NODE
:			
*			
** NCPB SWITCHED SDLC CONNECTION			
*			
NCPBSW	GROUP	LNCTL=SDLC,	SDLC LINE CONTROL
		DIAL=YES	SWITCHED CONNECTION
TONCPB	LINE		
	PU		
:			

Type 2.1 Casual Connection

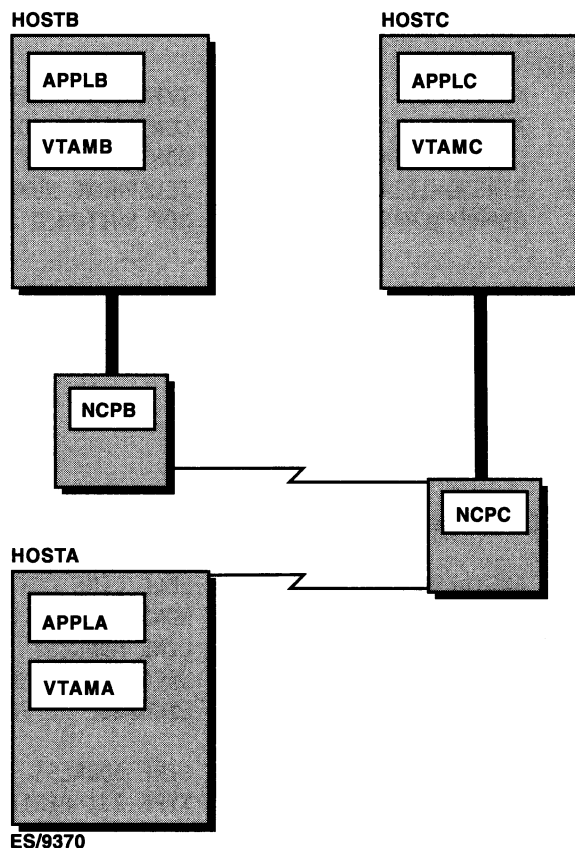


Figure 105. NCP Type 2.1 Casual Connection

VTAMA (the ES/9000 series processor) is defined to VTAMC and NCPC as a type 2.1 peripheral node using a PU definition statement following the LINE definition statement in NCPC that defines the nonswitched SDLC connection. On the link in VTAMA, the MODE operand indicates that NCPC is the secondary SDLC station and that VTAMA is the primary SDLC station.

The other node (VTAMB with NCPB) is defined to VTAMC with NCPC as a type 2.1 peripheral node using a PU definition statement in the switched major node because of the type of connection used between the two peripheral nodes.

The application programs in VTAMA and VTAMB can send BINDs to and receive BINDs from the application program in VTAMC. APPLC is defined in both VTAMA and VTAMB as a cross-domain resource (CDRSC). The application programs (APPLA and APPLB) in these type 2.1 peripheral nodes (VTAMA and VTAMB) are also defined to VTAMC as independent logical units using CDRSC definition statements.

If the type 2.1 casual connection is switched, VTAM can establish the connection when a session request is received from the application program. That is, for a peripheral node connection, in contrast to a switched subarea node connection, VTAM initiates the dial-out operation when an application program requests a session. Therefore, in Figure 105, where the connection between NCPB and NCPC is a switched SDLC link, if APPLB requests a session with APPLC, VTAM initiates the dial-out operation with the NCP as a part of the session setup process.

Considerations for Using Casual Connection

A type 2.1 casual connection can be used instead of a subarea connection to establish peer-to-peer communication between two VTAMs or VTAM and NCP. However, you should consider several things when using this unique type of connectivity.

Any VTAMs or NCPs that communicate using a casual connection and that reside in different SNA networks require the non-native network connection function; this function must be in all hosts involved in the session setup. For information about connecting multiple networks, see Chapter 8, “Connecting Multiple Subarea Networks” on page 437.

A casual connection is limited to a single link connection. A subarea connection can utilize either a multiple link transmission group (TG) or parallel TGs, but a type 2.1 node casual connection cannot. Casual connection does not support virtual route pacing and transmission group thresholds because these are peripheral node connections. Only fixed or adaptive session pacing can be used as the flow control mechanism.

Because there is no CDRM session across a casual connection, all LU-LU sessions must be initiated by one of the LU session partners directly sending a BIND request. A BIND that traverses a casual connection link must be either an extended BIND or an LU 6.2 BIND. LU 6.2 applications are generally not constrained by these restrictions because a BIND is used to initiate a session. However, sessions between other logical unit types can be supported only if the PLU always initiates the session. To establish a cross-domain session for a dependent LU, the SLU usually routes the session request to the appropriate VTAM using the CDRM session. Because the CDRM session does not exist, the session cannot be established. The session setup can succeed only if an independent LU or a VTAM application, acting as the primary logical unit, sends the BIND without the secondary logical unit (SLU) initiating the request. A session with a VTAM application program can also be initiated using the VTAM automatic logon facility. However, the automatic logon (LOGAPPL or VARY LOGON) must be invoked from the VTAM host connected to the independent LU by a type 2.1 channel.

A VTAM application program acting as an SLU cannot request a session with another VTAM application program by using the REQSESS application program macroinstruction. VTAM application program macroinstructions that can be used to initiate sessions are SIMLOGON and OPNDST. An application program cannot use the following VTAM functions over a type 2.1 casual connection:

- The INQUIRE macroinstruction to determine the status of an application program in another domain because that requires a CDRM session
- The CLSDST OPTCD=PASS macroinstruction to initiate a session between the SLU and a different application program (PLU) before terminating the session between the SLU and the current application program
- The RELRQ option on the SIMLOGON macroinstruction and the RELREQ exit routine for an application program to request a session with a logical unit that is currently in session with another application program.

Some of the various NetView-to-NetView sessions (such as OST-NNT) that are used for operator commands and message forwarding cannot be established between VTAMs that are using a casual connection. NetView uses one of the

restricted VTAM application program interface macroinstructions (REQSESS) to initiate these sessions. Therefore, when you use casual connection, you cannot use the following NetView facilities:

- Message forwarding
- Message automation based on message forwarding
- Operator command routing
- Terminal access facility (TAF) sessions, including the ability to browse messages and view status monitor panels in another domain
- Alert forwarding for NetView Release 2.

The following NetView facilities are available:

- Session monitor-to-session monitor communication
- Hardware monitor-to-hardware monitor communication, if both NetView programs are NetView Release 3.

In summary, if your environment meets the following requirements, you can use a casual connection for VTAM and NCP type 2.1 peer-to-peer communication:

- All session requirements can be satisfied by the primary logical unit's initiating the session or by your using automatic logon facilities; no secondary LU requests are supported.
- No application programs (IBM, other vendor-supplied, or user) use the restricted VTAM application program macroinstructions.
- Multiple and parallel link transmission groups are not required.

For example, if your environment meets these requirements, you can use casual connection between two subarea nodes for periodic communication over switched SDLC links for some file transfer operations.

Routing in a Subarea Network

This section contains the following subsections:

- How Session Traffic Is Assigned to a Specific Route
- How to Plan Routes in Your Network
- Subarea Addressing.

How Session Traffic Is Assigned to a Specific Route

In a subarea network, data traffic for a session is assigned to a virtual route. The virtual route then maps to an explicit route over which data traffic flows at a designated priority for the session. Following are details of how session traffic is assigned to a specific route.

Logon Mode Table

Secondary logical units specify a logon mode entry in an LU-LU session-initiation request. Classes of service that you want to be used for an LU-LU session are specified in logon mode table entries using class-of-service (COS) table names similar to the following:

```
LOGMODE1 MODEENT COS=BATCH,...
LOGMODE2 MODEENT COS=INTERACT,...
:
```

If both session endpoints reside in the same subarea, the class of service requested by the secondary logical unit is ignored because data flowing between the two endpoints never enters the network. Otherwise, VTAM obtains the requested class of service from the specified MODEENT macroinstruction in the logon mode table associated with the secondary logical unit. VTAM then uses the requested class of service to map to the class-of-service table.

Notes:

1. If the specified logon mode entry does not exist in the logon mode table associated with the secondary logical unit, the default logon mode (ISTCOSDF) is used if it is found in the table and its use is allowed as determined by the ISTCOSDF start option.
2. If no logon mode table has been designated for a logical unit, VTAM uses the IBM-supplied logon mode table, ISTINCLM. For details on ISTINCLM for your operating system, see "Logon Mode Table" in the *VTAM Resource Definition Reference*.
3. If a logon mode table entry specifies a class-of-service name that is not defined in the class-of-service table, VTAM rejects the session-initiation request.

Class of Service (COS) Table

A class of service specifies a set of performance characteristics used in routing data between two subareas. Different classes of service can be defined in a network for different types of users and data. For example, interactive sessions can be assigned to a fast route, batch jobs to a high-bandwidth route, and sessions involving the transmission of sensitive data to a secure route. At least one class of service can also be specified for SSCP sessions.

Different classes of service are specified in a COS table. Each class of service is specified by a symbolic name (for example, INTERACT for interactive sessions or BATCH for batch sessions). The COS table lists, in order of preference, the virtual route and transmission priority pairs that are to be used for each named class of service. For example:

```
INTERACT COS VR=(1,1),(0,1),...
BATCH    COS VR=(0,0),(1,0),...
:
```

When the logon mode table entry for an LU-LU session does not contain a COS name, VTAM uses the unnamed COS entry. The unnamed COS entry is one of two special entries that you should include when you code a COS table. The other is ISTVTCOS.

ISTVTCOS

ISTVTCOS is the special COS entry you use to specify the routes you want to use for SSCP sessions (SSCP-SSCP, SSCP-PU, and SSCP-LU).

VTAM looks for and uses the ISTVTCOS entry any time there is an SSCP session request for another subarea. If the

ISTVTCOS entry is not present, the unnamed COS entry is used.

Unnamed Entry

An unnamed entry (the entry name consists of eight blanks) can be placed in the COS table and is used when either of the following is true:

- No class-of-service name is obtained from the logon mode entry for an LU-LU session.
- No ISTVTCOS entry exists in the COS table, and an SSCP session has been requested.

If the unnamed class-of-service entry is requested but has not been included in the COS table, VTAM uses its own class-of-service defaults.

You need not define a COS table if the only COS names to be used are ISTVTCOS and the unnamed class of service; VTAM uses its own class-of-service defaults.

IBM-Specified Class-of-Service Defaults

VTAM has a default list of virtual route and transmission priority pairs that is used when either of the following is true:

- No COS table has been created.
- A COS table has been created, but:
 - No class-of-service name has been obtained from the logon mode entry associated with an LU-LU session request, and no unnamed entry has been included in the COS table.
 - An SSCP session has been requested, and no ISTVTCOS entry or unnamed entry has been found in the COS table.

The defaults consist of all 24 possible virtual route and transmission priority pairs in the following order:

```
VR0,TP0; VR1,TP0; VR2,TP0; ... VR7,TP0;  
VR0,TP1; VR1,TP1; VR2,TP1; ... VR7,TP1;  
VR0,TP2; VR1,TP2; VR2,TP2; ... VR7,TP2;
```

This default list might not be the best for your needs, and its use might not provide optimal network performance. You can replace the default list by creating a COS table with an unnamed COS entry containing the new list. This new default list is then used if no COS entry is named in an LU-LU session request or if an SSCP session is requested and no ISTVTCOS entry exists in the COS table.

Substituting Class-of-Service Parameters

Usually, if VTAM receives a user session request using a COS name that is unknown to this VTAM, the session is rejected. If you want the user session to be established despite the unknown COS name, you can enable VTAM to accept the unknown COS name and use substitute parameters that are coded on another COS name. To do this, code SUBSTUT=YES on the COS name that will provide the substitute parameters.

For example, in Figure 106 on page 417 assume that you are trying to set up a session between APPLA and APPLB, and APPLB is the

primary LU. VTAMA specifies COSA as the COS to be used for the session. VTAMB does not have COSA coded, so VTAMB uses the parameters from COSB to set up the session because COSB has SUBSTUT=YES coded. If you display the session from either VTAMA or VTAMB, the display will indicate that COSA is being used, even though the parameters for COSB are being used. The substitution is transparent to VTAMA. However, you can monitor the substitution in VTAMB and reject session establishments that you do not want to use the substitute parameters by coding a session management exit routine. For information about coding the exit routine, see “Session Management Exit Routine” in *VTAM Customization*.

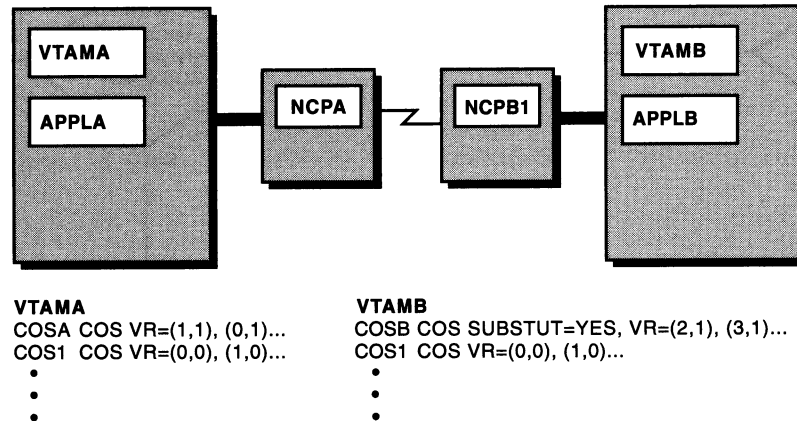


Figure 106. Class-of-Service Substitution

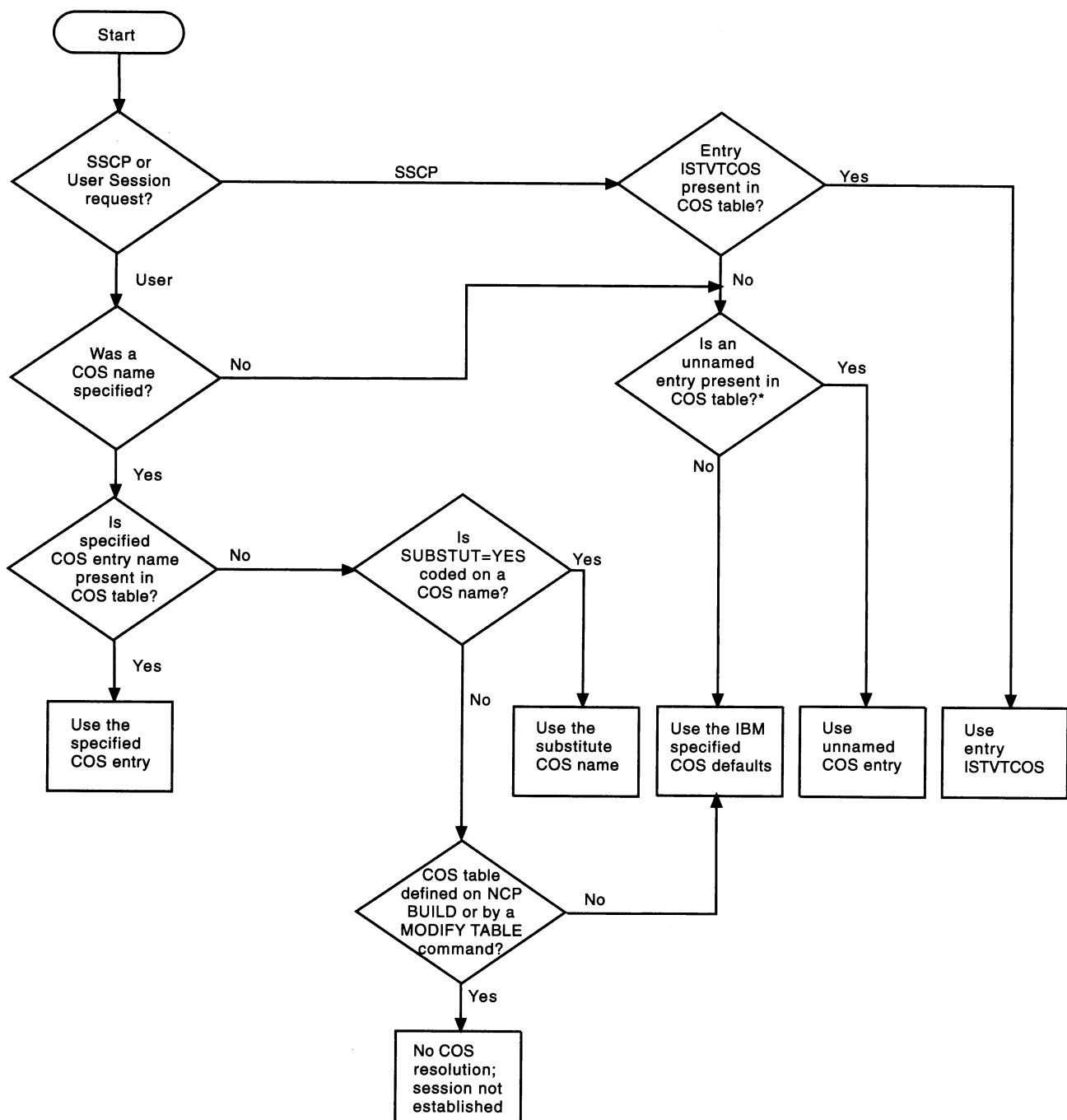
Substitute class-of-service parameters are not passed between VTAMs. If a request is forwarded to an adjacent SSCP, the COS name that was originally requested is the name passed. The substitute COS name is not passed.

The class-of-service hierarchy in Figure 107 on page 418 summarizes the steps VTAM takes to determine the appropriate COS table entry to use for SSCP and user sessions (or to determine to use the IBM-specified class-of-service defaults).

The SSCP selects the first virtual route and transmission priority pair listed for the requested class of service (or the first pair listed in the IBM-specified class-of-service defaults) that has been defined to the subarea of the session partner. An attempt is made to activate the virtual route.

The virtual route maps to an explicit route over which data traffic flows at the designated priority for the session. If all the network resources along the explicit route are operative, the virtual route is activated and available for session use.

If the explicit route is not operative, the next virtual route and transmission priority pair for the class of service requested is selected, and an attempt is made to activate that virtual route. If all virtual route and transmission priority pairs for the class of service requested map to inoperative explicit routes, the request to establish a session is either queued or rejected. Requests for SSCP sessions are queued (if queuing is allowed), but all others are rejected.



*This is also referred to as a "blank entry."

Figure 107. Class-of-Service Hierarchy

How to Plan Routes in Your Network

Designing routes through a network ranges in complexity depending on the number of subareas, connections between subareas, and types of user sessions that your network supports. Also, decisions must be made to achieve some combination of the following objectives, which involve compromises:

- Maximize quantity of data transmitted
- Maximize data security
- Maximize route availability

- Minimize transmission time
- Minimize cost
- Minimize data loss or the necessity to retransmit data
- Minimize traffic congestion.

NETDA/2 is a tool that can help plan for routes in your network and other network design tasks, such as:

- Designing a network
- Producing reports on designs so that you can evaluate them on performance
- Generating route statements.

The following steps can also help you make the decisions necessary to plan the routes in your network:

1. Determine the classes of service that will exist in your network. The following questions are the types of questions you need to answer in determining the classes of service you need.
 - Will you have one class of service for all interactive sessions?
 - Do you want a class of service that provides quick response times suitable for high-priority interactive sessions and a class of service that provides response times suitable for low-priority interactive sessions?
 - Do you want a class of service that provides routes that have the best availability?
 - Do you want a class of service that is suitable for batch processing?
 - Do you want a class of service that is suitable for high-security transmissions?
2. Identify the possible explicit routes available in your network, including the possible ways parallel links between communication controller subarea nodes could be divided among transmission groups.

Dividing parallel links into multiple transmission groups creates multiple explicit routes between two nodes. Multiple explicit routes between two nodes provides the opportunity to physically separate the data traffic for different classes of service in your network.

Combining parallel links between communication controller subarea nodes into a single transmission group increases the probability that explicit routes using the transmission group will be operational. A transmission group that consists of parallel links will still be available even if one of the links fails, and data traffic can continue to flow on the remaining links. Only links that have similar transmission characteristics (for example, high transmission speed) are normally placed in the same transmission group.

3. Analyze the physical characteristics of each identifiable explicit route, such as:
 - Length
 - Transmission time
 - Lowest link capacity in the route
 - Lowest transmission group capacity in the route
 - Lowest security level in the route
 - Operational probability.
4. Match the classes of service you selected to explicit routes having appropriate characteristics. Remember to consider the characteristics of a route between

each individual subarea pair. For example, for a class of service requiring a fast route, choose the fastest route between each subarea pair along the route.

Also, be sure to consider alternate routes for each class of service. If a subarea node or transmission group fails, all routes passing through that node or transmission group become inoperative and existing sessions are disrupted. An alternate route is a route that is available if the route being used becomes inoperable. If you want to ensure that you always have a route available, you should choose alternate routes that do not pass through the same subarea nodes or transmission groups as the routes that they back up.

5. Identify a virtual route for each selected explicit route.
6. Create an ordered set of virtual route and transmission priority pairs for each class of service.

In creating this ordered set, you assign a transmission priority to a virtual route which, together with the physical characteristics of the explicit route, determines the suitability of a particular virtual route for a designated class of service. Remember that multiple sessions can use the same virtual route (with the same or a different class of service).

For example, for a class of service requiring faster data transmission and predictable response times (a class of service suitable for interactive sessions), you can list virtual routes assigned to explicit routes that have fewer physical elements before listing virtual routes assigned to longer explicit routes. The shorter routes would be used first, and the longer routes would be used only for backup purposes.

However, you might have interactive end users with differing class-of-service requirements. The virtual routes for each of these classes of service could be identical to those above, and the class of service would be realized through the transmission priority. For the interactive users requiring faster data transmission and predictable response times (for example, an inquiry-response class of service), you would list virtual routes having a higher transmission priority than the virtual routes for interactive users where slower data flow is acceptable (for example, a data-collection class of service). In other words, sessions for different kinds of applications can be in progress over a given explicit route, and transmission priorities determine the order of transmission.

Suppose you have two classes of service for low priority and higher priority batch sessions. The higher priority batch sessions would be assigned to virtual routes with a higher transmission priority than the virtual routes for the lower priority batch sessions. For both batch classes of service, you would probably list high-bandwidth explicit routes before listing the explicit routes used for interactive sessions. The shorter routes used for interactive sessions could be used as alternate routes for batch sessions. The batch sessions would likely be assigned a transmission priority lower than the transmission priority of the interactive sessions using the same virtual route.

Note: In migration environments, you should include VR 0, TP 0 in each COS table entry as an identifier for default routes containing non-extended addressing nodes.

7. For details on coding a COS table to hold the ordered virtual route and transmission priority pairs for each class of service, see "VR" in "Class of Service (COS)" in the *VTAM Resource Definition Reference*.

8. As discussed in “How Session Traffic Is Assigned to a Specific Route” on page 414, VTAM processes the virtual route and transmission priority pairs in a COS table entry sequentially, starting with the first entry. If you require a more precise selection algorithm, use the session management exit routine or virtual route selection exit routine to change the virtual route selection process for LU-LU sessions.

For example, if you provide a virtual route selection exit routine, your routine receives the ordered virtual route list as a parameter from VTAM. Your routine can modify the list. VTAM then uses your reordered list of virtual routes to select a virtual route for the session.

If you've provided a virtual route selection exit routine, VTAM calls it whenever a session between a primary logical unit in the VTAM subarea and a logical unit in another subarea is about to be established. The virtual route selection exit routine is not called if both logical units are in the same VTAM subarea.

For more information about coding a virtual route selection exit routine or using the session management exit routine to change the virtual route selection process for LU-LU sessions, see “Virtual Route Selection Exit Routine” or “Session Management Exit Routine” in *VTAM Customization*.

9. For each subarea node, use PATH definition statements to create the path definitions necessary for defining the explicit routes and virtual routes that exist between the subarea node and other subareas. For example, the PATH statements needed in HOSTA of Figure 40 on page 155 follow:

```
PATH1  PATH DESTSA=NCPA1,
        ER0=(NCPA1,1),
        ER1=(NCPA1,1),
        VR0=0,
        VR1=1
PATH2  PATH DESTSA=NCPA2,
        ER0=(NCPA1,1),
        ER1=(NCPA1,1),
        VR0=0,
        VR2=1
```

From path definitions like these, VTAM builds a routing table for a subarea node similar to the table for HOSTA shown in Figure 39 on page 153. In creating your path definitions, list virtual routes only in those subareas that are going to activate them. For NCP subareas that cannot activate virtual routes, the virtual routes ending in those subareas are dynamically defined to those subareas as part of route activation.

Coding paths for a multiple-domain environment is similar to coding paths in a single-domain environment. Communication between domains requires paths to be defined between VTAM subareas. The PATH definition statements that are used to define routes between VTAM and NCP subareas in a network are also used to define routes between the VTAM domains. For each subarea, the PATH definition statements define the explicit and virtual routes that connect the various VTAM domains.

VTAM can function as an intermediate routing node. VTAM receives the data from the adjacent subarea and transmits it to the next subarea on the network route as defined by the PATH definition statements. The IRNSTRGE start option is used to define the maximum size of the virtual area in VTAM storage that can save host intermediate routing node (IRN) transmissions. If IRNSTRGE=0 (the default) is used, the amount of storage is unlimited.

For further details on PATH statements or the IRNSTRGE start option, see the *VTAM Resource Definition Reference*.

10. One or more path definition sets must be activated for VTAM to know how to route cross-subarea session traffic. A path definition set is activated with a VARY ACT command. When a file containing a set of PATH definition statements is activated, it defines explicit routes and virtual routes to VTAM. At that point, the routes are defined but not yet active.

You can activate additional path definition sets to add or modify route definitions when necessary. You can define new routes and redefine or delete inactive routes. Existing route definitions not affected by a given path definition set activation are unmodified and remain in effect. VTAM rejects any attempt to modify the definition of any previously defined route that is not in the inactive state.

A path definition set is not a major node and cannot be deactivated.

11. In addition to the activation of path definitions, all physical elements included in an explicit route (subarea nodes and transmission groups) must be activated before an explicit route is ready for activation.

An explicit route is operational when physical connectivity between the endpoints is established. Path information is exchanged between each subarea when the connection between the subareas is activated. A virtual route becomes available for activation after its corresponding explicit route is active.

VTAM automatically attempts to activate explicit and virtual routes when they are first needed for a session. A virtual route remains active until all sessions assigned to it have ended, at which time it is automatically deactivated. The deactivation of a virtual route has no effect on the status of its associated explicit route. Once active, an explicit route is never deactivated, but it can become inoperative due to hardware failure or deactivation of physical elements within the route.

12. You can use the DISPLAY ROUTE command to display information about the explicit routes and virtual routes between the host subarea and a given destination subarea. You can also use this command to test whether one or more explicit routes to a given destination are operative (see “Displaying and Testing Routes” on page 352).
13. Remember that the requested class of service for a session is obtained from the logon mode table associated with the secondary LU. For details on creating a logon mode table and associating LUs with your table, see “Logon Mode Table” in the *VTAM Resource Definition Reference*. If you do not create a logon mode table and also make the association between an LU and that table, VTAM obtains the LU's requested class of service from the IBM-supplied logon mode table, ISTINCLM.

Subarea Addressing

Each subarea node must have a unique address. The number of subarea addresses available is determined by the capability of the VTAM or NCP release to support a range of addresses. Depending on the software level, the addressing method can be one of the following:

- Non-extended network addressing
- Extended network addressing

- Extended subarea addressing.

Compatibility between Non-Extended and Extended Network Addressing Nodes

For a network containing non-extended addressing nodes, the MAXSUBA start option specifies the highest subarea value that can be used throughout the network. A MAXSUBA of 63, for example, defines a network with up to 63 subareas and 1024 elements in each subarea. Code the MAXSUBA start option in the start option list if you want VTAM to communicate with non-extended addressing nodes, or if the MAXSUBA operand is coded in an NCP that VTAM will communicate with.

Extended network addressing is a network addressing structure that increases the size of the subarea address up to 255 in conjunction with up to 32,768 elements. VTAM V4R1 extends host element addressing to 65,535 elements.

Non-extended addressing nodes can continue to use this structure. They can, however, participate in an extended network addressing network only when the following two conditions are met:

- A compatibility program temporary fix (PTF) must be installed on all non-extended addressing nodes that communicate with extended network addressing nodes.
- All nodes that communicate with non-extended addressing nodes must define a MAXSUBA. Extended network addressing nodes need this information to decode 16-bit network addresses. The value specified for MAXSUBA must be the same at all nodes.

In addition, the following restrictions apply when extended and non-extended addressing nodes coexist in the same network:

- VTAMs that are non-extended addressing nodes cannot communicate with any logical unit whose element address exceeds the maximum element number defined by the subarea/element split.
- VTAMs that are non-extended addressing nodes cannot communicate with any node whose subarea is greater than the MAXSUBA.

Figure 108 on page 425 illustrates an element address incompatibility. Domain 1 contains nodes that use extended network addressing. Domain 2 contains nodes that use non-extended addressing node addresses. All nodes have defined a MAXSUBA of 63. Although all of the subareas are within the MAXSUBA, some of the element addresses exceed the maximum, which is 1023. Following are two address constraints on this network.

- In Domain 1, the terminals T1 and T3 can communicate with any subarea because their element addresses (100) are less than 1024. T2 and T4, however, cannot communicate with host A40M because their element addresses (1500 and 2000) are greater than 1023.
- In Domain 2, T5 and T6 can communicate with either host. However, they cannot communicate with any application program in host A60M whose element address is greater than 1023.

These limitations can be overcome through SNA network interconnection (SNI), which can be used to place non-extended addressing nodes and extended network

addressing nodes into separate networks. See Chapter 8, “Connecting Multiple Subarea Networks ” on page 437 for information about implementing SNI.

Although it is not recommended, a network containing non-extended addressing nodes can have subareas whose subarea number is greater than the MAXSUBA. The non-extended addressing nodes are unable to communicate with these subareas. In this situation, the following constraints apply:

- A non-extended addressing node VTAM cannot appear in any route whose endpoints have a subarea address greater than MAXSUBA.
- A non-extended addressing node VTAM cannot be adjacent to a node whose subarea is greater than MAXSUBA.
- A non-extended addressing node NCP can be adjacent to a node whose subarea is greater than MAXSUBA, but only if the NCP is used as an intermediate routing node (IRN). This requires a PTF.

Table 26 and Table 27 show the adjacent and endpoint node subarea requirements.

Table 26. Adjacent Node Subarea Requirements for Multiple-Domain Environment

	Pre-ENA VTAM	ENA VTAM Subarea ≤ MAXSUBA	ENA VTAM Subarea > MAXSUBA
Pre-ENA NCP	Yes	Yes	Yes *
ENA VTAM Subarea ≤ MAXSUBA	Yes	Yes	Yes
ENA NCP Subarea > MAXSUBA	No	Yes	Yes

Note:

Yes These two nodes can be endpoints.
 No These two nodes cannot be endpoints.
 * Pre-ENA is used as an IRN only.

Table 27. Endpoint Node Subarea Requirements for Multiple-Domain Environment

	Pre-ENA VTAM	ENA VTAM Subarea ≤ MAXSUBA	ENA VTAM Subarea > MAXSUBA
Pre-ENA NCP	Yes	Yes	No
ENA VTAM Subarea ≤ MAXSUBA	Yes	Yes	Yes
ENA NCP Subarea > MAXSUBA	No	Yes	Yes

Note:

Yes These two nodes can be endpoints.
 No These two nodes cannot be endpoints.

Figure 108 on page 425 illustrates these constraints. The third domain contains extended network addressing nodes whose subarea addresses are greater than MAXSUBA.

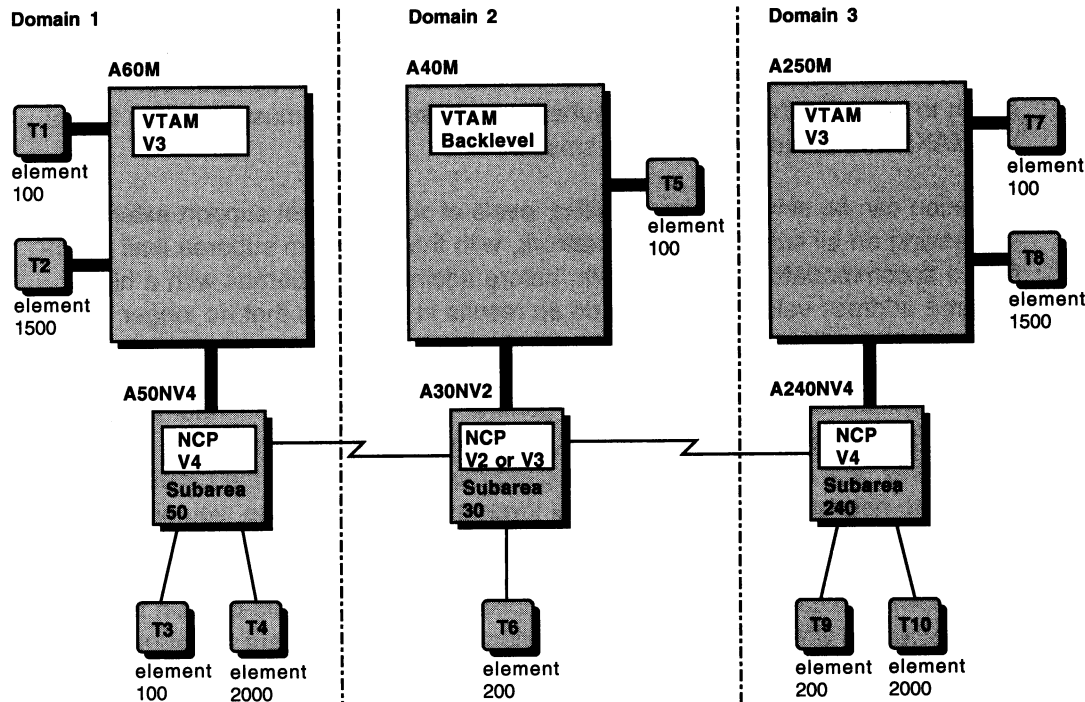


Figure 108. Element and Subarea Address Incompatibility in Multiple-Domain Environment

The following communication restrictions apply to the network in Figure 108 on page 425:

- None of the Domain 2 terminals can have a session with host A250M. Conversely, the Domain 3 terminals cannot have a session with host A40M.
- All of the terminals in Domain 3 can communicate with any application program in Domain 1, and all of the terminals in Domain 1 can communicate with any application program in Domain 3.
- NCP A30NV2 is a non-extended addressing node, but can still act as an IRN between Domains 1 and 3. However, the NCP's terminal, T6, cannot communicate with Domain 3.

As with the element address problem, subarea address incompatibilities can be avoided by using SNI to place nodes in separate networks. See Chapter 8, "Connecting Multiple Subarea Networks" on page 437 for information about implementing SNI.

Compatibility between Extended Network and Extended Subarea Addressing Nodes

Whereas extended network addressing supports a maximum subarea address of 255, extended subarea addressing allows you to specify subarea addresses higher than 255. Subarea addresses can be as high as 65535.

However, defining networks with large numbers of subareas can increase the storage requirements for VTAM and NCP. To control the maximum number of subareas in a network, a constant in the VTAM constants module and the NCP operand SALIMIT allow you to specify the maximum subarea size of the network. You can define VTAMs and NCPs in the network with varying values for the maximum subarea limit. However, the actual subarea addresses used must be within a common address range for the subarea nodes to communicate.

Processors and communication controllers with levels of VTAM and NCP that do not support extended subarea addressing can exist in the same network with other processors and communication controllers that do. For subarea node communication to be supported, the actual subarea addresses used must be within 255 or the MAXSUBA value, whichever is smaller.

Migration can be simplified by installing levels of software that support extended addressing on all subareas in the network, with the maximum subarea limit specified to accommodate planned growth, before adding any subareas with a higher subarea address value. Failure to do so results in subareas that no longer have connectivity to other parts of the network.

The maximum subarea limit should, in general, be constant throughout the network. If the subarea limit cannot be constant, you can try different values with no loss of connectivity until a subarea address is used that is higher than the lowest subarea limit in the network. When moving to a lower subarea limit value, you should lower all subarea numbers before changing the subarea limit value.

The following subarea addressing restrictions apply to a multiple-domain network unless you use SNA network interconnection (SNI) to circumvent the addressing constraints:

- Adjacent subareas can support different maximum subarea values and still communicate provided their actual subarea addresses fall within their common range. For example, a non-extended subarea addressing VTAM can communicate with an extended subarea addressing NCP that supports an SALIMIT value greater than 255 if the NCP's subarea address is no greater than 255.
- A subarea whose address is above the addressing supported by other adjacent subareas on a route cannot communicate to or through those subareas.
- Subareas can communicate through other subareas with an address higher than they support provided there is an intermediate subarea with an address acceptable to both. For example, a non-extended subarea addressing VTAM could communicate through an NCP with a subarea address of 300. There would, however, have to be an intermediate VTAM or NCP supporting extended subarea addressing with an address no greater than 255 and an SALIMIT of at least 300.
- An SSCP cannot activate any subareas with addresses greater than its addressing capabilities.
- For an extended subarea addressing NCP to successfully send a CONTACT request unit to an adjacent extended subarea addressing NCP with an address above 255, the link station must be owned by an extended subarea addressing VTAM.

With extended subarea addressing, VTAM and NCP support the use of 16 explicit routes. You can define up to 16 explicit routes for each destination subarea. However, for an explicit route to have a number greater than 7, all subareas on the route must support extended subarea addressing, and all NCPs must have ERLIMIT=16 coded on the BUILD definition statement.

Establishing and Controlling SNA Sessions

The session management exit routine can be invoked in the domains in which each of the session partners resides and can be used to select adjacent SSCPs for LU-LU session and resource status requests.

The adjacent SSCP selection function in the session management exit routine can be used to control the SSCP to which a session request is sent. The exit allows you to shorten or reorder the list of adjacent SSCPs. You can use this function in conjunction with the dynamic adjacent SSCP table facility so that you can avoid defining adjacent SSCP tables while maintaining control of session request routing.

Operating VTAM in a Multiple-Domain Subarea Network

This section contains VTAM operation considerations for a multiple domain subarea network. For more information about operating VTAM, see “Operating VTAM” on page 159.

Activating Resources

VM,VSE Connecting an NCP or another VTAM by an SDLC, LAN, or X.25 link through a communication adapter creates a cross-domain connection. Because that NCP is a cross-domain resource, it cannot be activated by the VTAM in the other domain nor can its link station (defined in the channel-attachment major node) be used as RNAME, LOADSTA, or DUMPSTA either on a PCCU definition statement or in a VTAM command.

VM,VSE There are restrictions on VTAM's support of channel-attached LAN, X.25, BSC, and SDLC lines when they are defined in a channel-attachment major node. Dynamic reconfiguration is supported only for PUs and LUs under channel-attached NCPs, or LUs under local SNA, packet, or SDLC non-switched major nodes. In addition, the facility to change line scheduling parameters is an NCP function and does not apply to BSC lines attached through the communication adapter.

In multiple-domain networks, ADJSSCP tables should be activated before cross-domain resources, unless the dynamic adjacent SSCP function is specified.

When activating cross-domain resource managers, first activate the host cross-domain resource manager in each domain. This enables each host cross-domain resource manager to request sessions with and accept session requests from other known cross-domain resource managers.

After you have activated the host cross-domain resource manager, you can then activate any CDRM major nodes that represent other domains. VTAM then requests a session with the other SSCP. The request is rejected unless the other domain has activated both its host cross-domain resource manager and the external cross-domain resource manager representing the SSCP making the request.

Requests for sessions with cross-domain resources that are not defined are defined dynamically by the cross-domain resource manager, if you have authorized dynamic definition of cross-domain resources.

For autologon sessions, if a session request cannot be routed to its destination because an SSCP-SSCP session has not been established, the session fails and the operator receives a notice of the failure. After the needed SSCP-SSCP session is active, the session request is retried.

To cancel the autologon specification, a VARY NOLOGON command can be issued in the originating logical unit's domain, or the logical unit can log on to a controlling LU. For non-autologon sessions, the operator receives notification of the failure, but the session attempt is not automatically retried.

Deactivating Resources

VTAM deactivates a cross-subarea link to a host processor when the channel-attachment major node is deactivated. Deactivating this link (a channel) destroys any sessions using a route on that link, possibly including the SSCP's session with another subarea node. Also, VTAM might be an intermediate routing node for cross-domain sessions that traverse the link. For this reason, care should be taken not to deactivate a cross-subarea link that could be supporting session traffic.

Note: When deactivating any channel-attachment major node, or any of its subordinate links or link stations, you might be deactivating a link that carries intermediate routing node traffic. Deactivating the link will disrupt these sessions.

When you enter a VARY INACT command for an NCP, VTAM does not deactivate any cross-domain link or link station unless CDLINK=INACT is specified on the VARY INACT command.

You can deactivate and reactivate CDRMs without disrupting active sessions. Use the SAVESESS operand on the VARY INACT command to specify whether active sessions should be terminated when you deactivate the CDRM. This operand enables you to keep active sessions when you use the FORCE or IMMED option of the VARY INACT command. However, SAVESESS specifies that all pending or queued sessions using the specified CDRM are terminated.

The loss of an SSCP-SSCP session (for example, as the result of a link failure) is not the same as external cross-domain resource manager deactivation and does not terminate existing LU-LU sessions. However, new sessions with logical units controlled by the SSCP that were lost cannot be started until the SSCP-SSCP session is re-established. For information (including migration considerations) on SSCP session recovery, see *Planning for NetView, NCP, and VTAM*.

Operating with an NCP

The domain of an NCP is determined by the owning VTAM (the VTAM that activated it). For example, in Figure 109 on page 430, NCP1 and NCP2 are activated by HOST1, and thus are in HOST1's domain. NCP3 has not been activated by HOST1, and is therefore not in HOST1's domain. Thus, the link between NCP1 and NCP2 is a same-domain link, and the link between NCP2 and NCP3 is a cross-domain link.

Links and Link Stations

In Figure 109 on page 430, if NCP2 is inactive and LINK1 is active, the operator at HOST1 can dump NCP2 by specifying ID=LINKSTA1 on the MODIFY DUMP command. If HOST2 activates NCP3, HOST2 can dump NCP2 by specifying ID=LINKSTA4.

Activating Links

The same-domain link between NCP1 and NCP2 in Figure 109 on page 430 is made available through the activation of LINK1 and LINKSTA1 by HOST1. This link and link station can be:

- Directly or indirectly activated. (For example, you can enter specific VARY ACT commands for LINK1 and LINKSTA1, or LINK1 and LINKSTA1 can be defined to be initially active in NCP1's definition statements.)
- Automatically activated when NCP2 is activated (by specifying LINKSTA1 on the RNAME operand of the VARY ACT command or on the RNAME operand in the PCCU definition statement for NCP2).

Note that in Figure 109 on page 430, the activation of LINK2 and LINKSTA2 by HOST1 has no effect on the availability of the link between NCP1 and NCP2. The activation of LINK2 and LINKSTA2, which is required to make the link itself available, must occur by SDLC monitor mode being specified for LINK2 in NCP2's definition, or by HOST2 activating NCP2, LINK2, and LINKSTA2. If LINK2 and LINKSTA2 are activated by HOST1, HOST1's VTAM does not deactivate them when NCP2 is deactivated. In this case, VTAM loses its ownership of LINK2 and LINKSTA2 within NCP2 by virtue of the deactivation command sent to NCP2 itself.

Deactivating Links

If LINK1 and LINKSTA1 are active when NCP2 is deactivated, LINK1 and LINKSTA1 are automatically deactivated, unless you directly or indirectly activate them.

Therefore, if the link between NCP1 and NCP2 continues to carry session traffic after NCP2 is deactivated by HOST1, you should ensure that LINK1 and LINKSTA1 are directly or indirectly activated. This can be verified by entering a DISPLAY command for the link or link station (a DISPLAY of either resource displays the status of both). As a result, the only path available for session traffic to flow using the link between NCP1 and NCP2, given that HOST1 is deactivated, is from NCP1 to NCP2 to NCP3 to HOST2 for PLUs in the host. If PU type 2.1 nodes are attached to NCP1, NCP2, or NCP3, that link can also be used for session traffic.

The order in which the NCPs are deactivated determines how a VTAM domain contracts. The deactivation of an NCP by a VTAM host removes that NCP from the domain of that host. After an NCP is deactivated in a domain, all links leading to it are considered cross-domain links. During an NCP deactivation, then, same-domain links might become cross-domain links. If two or more NCPs are deactivated simultaneously (except as a result of a HALT command; see "Halting VTAM" on page 356), the time when the change occurs is unpredictable, and it might be difficult to apply the previously described rules for not disrupting sessions. Unless these rules are followed so all links have the same final disposition (active or inactive), regardless of which rule applies, do not deactivate the NCPs simultaneously.

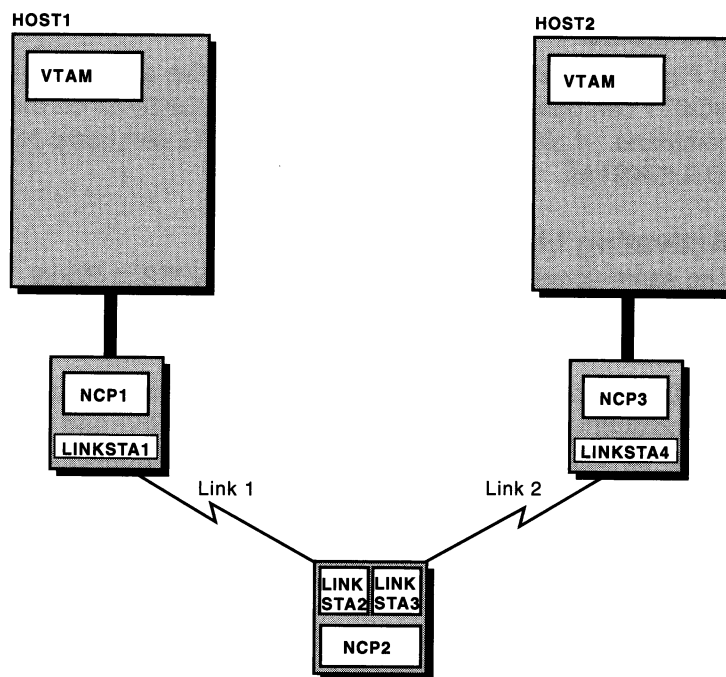


Figure 109. Effects of NCP Deactivation on Cross-Subarea Links and Link Stations

Cross-Domain Links

The cross-domain link between NCP2 and NCP3 is made available through the activation of LINK3 and LINKSTA3 by HOST1. Because this link is a cross-domain link, by definition, NCP3 is not activated by HOST1, and LINK3 and LINKSTA3 can not be automatically activated. Therefore, a cross-domain link and link station must be directly or indirectly activated. When NCP2 is deactivated and CDLINK=INACT is specified in the VARY INACT command, VTAM indirectly deactivates all cross-domain links and link stations (because they are below the NCP in the resource hierarchy). LINK2 and LINKSTA4 must have become active in NCP3 either through the SDLC monitor mode function or through activation by HOST2.

Delayed Activation of Logical Lines

This function enables you to perform an orderly activation of a physical resource and its associated logical resources. Because there is a hierarchical relationship between the physical resources (line and physical unit) and the logical lines, you can use this function to implicitly activate a logical line when the physical resources are explicitly activated.

The function creates a hierarchy between a logical line in a group (within an NCP major node) that has a PHYSRSC operand coded and the physical resource indicated by the PHYSRSC operand. This includes:

- NCP/Token-Ring interconnection (NTRI) physical resources (line and physical unit) and logical lines for INN link stations
- NCP Packet Switched Interface (NPSI) physical resources (line and physical unit) and logical lines
- Frame-relay physical resources (line and physical unit) and logical lines.

If the OWNER operand is specified for both the physical unit and the logical line, the logical line is subordinate to the physical unit only when the owner of the phys-

ical unit is an owner of the logical line and the PHYSRSC operand is coded for the logical line group. Remember that a physical unit can have only one owner, but a logical line can have multiple owners. If the OWNER operand is not coded for both the higher-level physical unit and the logical line, the logical line is always hierarchically subordinate to the physical unit, if this VTAM can be the owner of both resources. (For information about the OWNER operand, see “OWNER” in the *VTAM Resource Definition Reference*.)

Because only one VTAM can own the physical unit, yet the physical unit can be built into the hierarchy in multiple VTAMs (unless the OWNER operand is specified), only one VTAM can have the physical unit active and have logical lines subordinate to the physical unit. When explicit owners are not specified, if another VTAM owns a higher-level physical unit and the physical unit is activated by a second VTAM, the activation request issued by the second VTAM is rejected by the NCP. In this case, the second VTAM removes the higher-level physical unit from the hierarchy of all its subordinate logical lines. In this way, the logical lines within the second VTAM can still be activated by a mechanism other than being subordinate to the physical unit. After the hierarchy is broken in this way, it is not reconstructed even if this VTAM successfully reactivates the physical unit at a later time.

For most multiple-domain environments, the OWNER operand should be specified for both the physical unit and the logical lines so that the owner of the physical unit and the set of subordinate resources is predetermined and does not vary. The OWNER operand provides a way to control activation while also providing a backup owner for the physical resource. This relationship remains even if there is a take-over of the NCP. For example, if the VTAM that owns the physical unit (and possibly some logical lines) releases the NCP and another VTAM acquires the physical unit, the physical unit is placed hierarchically above any logical line acquired from the other VTAM.

Activating logical lines hierarchically

Activation processing in a multiple-domain environment is the same as in a single-domain environment except for the difference in hierarchical relationships that could exist because of the checking of the OWNER operand. When a logical line is subordinate to a physical unit, the physical unit must be activated before the logical line is activated. When the physical unit is owned by another VTAM, but this VTAM is to own a logical line subordinate to the physical unit, the activation of the physical unit must be attempted so that this VTAM can see that the physical unit is owned by another VTAM. Note that a logical line that specifies a different OWNER from its associated physical resource is not hierarchically subordinate to that physical unit. In this case, the activation of the physical unit need not be attempted before the logical line can be activated.

Deactivating logical lines hierarchically

Deactivation processing in a multiple-domain environment is the same as in a single-domain environment except for the difference in hierarchical relationships that could exist because of the checking of the OWNER operand. In addition, hierarchical relationships could be affected by the rejection of a activation request sent by a VTAM to a higher-level physical unit that is owned by another VTAM.

Discontiguous Domains

Because the connectivity of a domain changes when an NCP is deactivated, you should be aware of the possibility of creating a discontiguous domain. In Figure 109 on page 430, for example, if HOST1 had activated NCP3 as well as NCP1 and NCP2, and if HOST1 then deactivated NCP2, NCP3 would be in an isolated part of HOST1's domain. That is, HOST1, NCP1, and NCP3 would all be part of the domain, but the only access to NCP3 from HOST1 would be through NCP2, which would not be in the domain. Continued session traffic might not be possible to resources in or beyond NCP3, depending on whether the deactivation of NCP2 caused deactivation of the cross-subarea links connected to NCP2.

Discontiguous domains should be avoided. As shown in Figure 109 on page 430, because HOST1 does not own the route to NCP3, the HOST1 operator does not get notification of failures of links and link stations in NCP2. However, the HOST1 operator always gets notification of explicit route failures. In addition, VTAM's ability to recover link station failures in NCP2 is lost. Finally, if NCP3 fails, the HOST1 operator does not have the capability of reloading that communication controller.

To avoid a discontiguous domain, always first deactivate those NCPs that are farthest away from the host, working inward toward the host. In this example, HOST1 should deactivate NCP3 first, NCP2 second, and NCP1 last. NCPs need not be deactivated before halting VTAM to ensure that the proper order of deactivation is followed. VTAM follows special procedures during its halt processing to simultaneously deactivate all NCPs that are still active in the domain (see "Halting VTAM" on page 356).

If a discontiguous domain exists, all cross-subarea links connected to an isolated NCP are cross-domain links.

Note: If you are implementing an SDLC switched connection between NCP subareas in a multiple-domain network, the link station in each NCP can be controlled by VTAM, and you can implement call security verification. For information about implementing call security verification, refer to "Switched SDLC Subarea Connection" on page 269.

Backing Up Resource Owners

Every resource in a network must be owned by an SSCP in some domain. When a host fails, another SSCP can, with the proper planning and definitions, assume ownership of the failing host's resources. Any host can take over the resources of any NCP with which it has connectivity, and the takeover can be nondisruptive for sessions.

Steps for Resource Takeover

The steps follow:

1. NCP and the physical and logical units that it is to back up, including resources defined in any dynamic reconfiguration files, should be already defined to the backup host.
2. The operator for the backup host issues the appropriate commands to acquire and activate the appropriate resources. This results in the takeover host activating the resources.

3. For dependent LUs, the backup host redefines to itself the CDRSCs it is backing up as LUs as each LU's PU becomes active. Each statically-defined CDRSC definition representing a dependent LU in the failing domain becomes a shadow resource definition. Independent LUs continue to be defined as CDRSCs but are now displayable under the PU that is serving as their adjacent link station, if an independent LU has an active LU-LU session over that adjacent link station.

Note: There are situations that cause a dependent LU to remain as a shadow resource following its PU becoming active. In those situations the LU is brought out of shadow once the LU-LU session is terminated.

4. CDRMs that have cross-domain resource definitions for the resources taken over by the backup host have these resources associated with the wrong owning CDRM name. Operators in these domains should use the MODIFY CDRM command to re-associate these CDRSCs with the new CDRM so that session-initiation requests can be routed to the proper domain or the CDRSC definitions in those domains can specify the VFYOWNER operand in conjunction with an ADJSSCP table to locate the CDRSC.

Restarting the Host

To restart a host, both of the following must occur:

- VTAM's representation of the network must be restored. **MVS,VSE** Using configuration restart facilities, you can have it restored to the state it was in before the host failed. NetView command lists can be used to automate the restarting of the network resources.
- Sessions must be re-established. Some sessions can continue despite the failure. For example, sessions between devices in this domain and application programs in other domains that have been defined to continue after automatic network shutdown of the NCP can continue. Of these, some sessions also continue when VTAM is started. If the device in session supports recovery, the session continues when VTAM reactivates physical units and LUs. If not, the session is disrupted.

Note: If the session is established with a non-extended BIND, the restarted VTAM loses knowledge of all sessions. Therefore, you should plan for the following two situations:

- Cross-domain sessions with peripheral LUs owned by the failing host can continue when VTAM fails, but without VTAM support. Those sessions that continue when VTAM restarts do so with limited VTAM support. That is, the DISPLAY operator command, when issued from the restarted host, provides only limited session information. Unformatted LOGOFF requests from these LUs must be specified as TYPE(FORCE) or TYPE(UNCOND) and must not refer to the session partner; that is, the APPLID operand must be omitted. Formatted Terminate Self requests from these peripheral LUs must be specified with the type code set to cleanup. TYPE=UNCOND or TYPE=FORCE must be specified in VARY TERM operator commands.
- Sessions that do not continue after VTAM failure must be re-established by the operator or one of the session partners, as necessary.

Returning Ownership

When the original resource owner is restarted, the backup host can release ownership of the resources using the GIVEBACK operand on the VARY INACT or VARY REL operator command. The original owning host can then resume ownership of the resources using the VARY ACT or VARY ACQ operator command. The operations of assuming ownership in the backup host, releasing ownership of the resources in the backup host, and resuming ownership of the resources in the restarted original host are all nondisruptive to the following session types:

- Nonswitched SDLC connections
- Switched SDLC connections
- Token-ring connections
- X.25 connections using permanent or switched virtual circuits.

However, each of the devices attached using these connections must be capable of supporting the activation recovery.

Note: If you specify the DELAY option on the ANS operand in NCP, the disruption of binary synchronous (BSC) sessions can be delayed when the original host fails. However, the activation of the BSC resources by the backup host results in the disruption of these sessions.

Sharing an NCP

An NCP can be shared by more than one SSCP. A VARY ACT command for an NCP causes the VTAM SSCP to activate the NCP, establishing an SSCP-PU session with the NCP.

Owning resources

If multiple VTAMs activate an NCP, each of those VTAMs establishes an SSCP-PU session with the NCP, and each is considered an owner of that NCP. Each VTAM can also activate and share ownership of the NCP's links and link stations. However, the NCP's physical units and logical units can belong to only one VTAM at a time. The VTAM that activates a physical unit first becomes the exclusive owner of that physical unit; the other VTAM cannot activate that physical unit until the SSCP that currently owns the physical unit deactivates it.

Defining owners

The resources of an NCP that cannot be shared can be explicitly divided among the VTAMs that own the NCP. You can use the OWNER operand in NCP definition statements to explicitly identify the VTAM that is to own a particular resource. If a resource is allocated to a particular VTAM by the OWNER operand, that resource can be controlled only by operator commands entered from that VTAM. Therefore, VARY ACT and VARY INACT commands can be entered for these resources from the VTAM that owns them, but not from any of the other VTAMs that share the NCP. The OWNER operand can be specified on the PCCU, GROUP, LINE, and PU definition statements.

Backup owner

You can also specify that the NCP's resources can be taken over by a backup VTAM by using the BACKUP operand in the NCP definition. The operator at a backup VTAM can enter a VARY ACQ command to acquire resources that belong to another VTAM. The VARY ACQ command can be used to acquire all of the NCP's resources, regardless of how the OWNER operand is specified, or can be used selectively to

acquire only resources assigned to a particular VTAM through the OWNER operand. You can return the resources by using the VARY REL command.

Dump or IPL of a shared NCP

You can automatically dump or IPL the same NCP from more than one channel-attached VTAM. (You can code YES for AUTODMP and AUTOIPL on multiple VTAMs for the same NCP.) If the NCP is already being IPLed or dumped by one VTAM and it receives a request for an IPL or dump from another VTAM, the second request is rejected and the operator is informed that the second VTAM is waiting for the communications controller to become available. When the contending operation completes, VTAM restarts the requested IPL or dump operation if the state of the communications controller indicates that it is required.

Note: More than one host might take a dump successfully.

This function requires SSP Version 3 Release 7 or a subsequent release.

Verifying a Multiple-Domain Subarea Network

This section suggests steps to follow to verify that your multiple-domain, subarea network functions properly. However, before you perform the steps in this section, you must perform the steps in “Verifying a VTAM Network” on page 160 for each domain in your network. After you have done this, make a list of the most important normal and backup operations in your network, and verify each of them one at a time. Then verify the network over a length of time, as in normal operation.

For specific information about how to perform these steps and for the syntax of operator commands, refer to *VTAM Operation*. If these steps do not yield the expected results, double-check the definition statements for the resources that are affected.

To verify that your multiple-domain, subarea network functions properly:

1. Activate cross-domain paths and links, cross-domain resource managers (CDRMs), and cross-domain resources (CDRSCs).
2. Verify cross-domain sessions by exchanging requests across domains, between terminals and hosts, and between application programs.
3. Verify the NetView program, if it is installed.
4. Verify host backup using a channel-attached, shared NCP.
5. Verify host backup over a cross-domain link.
6. Verify automatic SSCP-SSCP session restart.
7. Halt VTAM in each domain.

Chapter 8. Connecting Multiple Subarea Networks (MVS, VM)

This chapter describes a multiple-network environment that uses SNA network interconnection (SNI). You should be familiar with previous chapters before using this chapter.

Implementing a Multiple-Network Environment

A multiple-network environment consists of multiple independent SNA networks that are interconnected. The SNA network interconnection (SNI) facility enables communication between these separate networks.

SNA Network Interconnection

SNA network interconnection enables you to:

- Interconnect multiple, independent SNA networks so that terminal users in one SNA network can access information (application programs) in other SNA networks.
- Divide an existing network into smaller, independent networks, with communication between these networks maintained by the SNI function.

Following are descriptions of environments in which SNI can be used. (An enterprise is a business organization.)

Independent Enterprises: If two or more independent enterprises need to access and exchange information using their current SNA telecommunication networks, SNI can be used to interconnect these autonomous networks, which might have dissimilar characteristics.

Enterprise Mergers and Acquisitions: If two enterprises have merged or if one enterprise has acquired another, and the dissimilar SNA networks need to interconnect, SNI can be used so that any terminal or application program in one network can access any terminal or application program in the other network.

Multiple Networks within an Enterprise: If multiple SNA networks have evolved within a large enterprise, each with different network characteristics, SNI can be used to merge these dissimilar networks into a single logical network, yet maintain the autonomy of each.

In each of the preceding examples, SNI can be used to interconnect networks with dissimilar characteristics while preserving the independence of each participating network. These participating networks do not have to change to a common address structure or resource naming convention, and each network can preserve its existing management procedures and controls.

Defining Your Network

Defining your multiple-network environment involves identifying your different networks and defining the resources that will help you communicate between the networks. You need the following for controlling cross-network sessions:

Gateway VTAMs

Enable at least one gateway VTAM for each network gateway. The gateway VTAM controls sessions across the network gateway. The gateway VTAM is in session with the gateway NCP, assisting in address translation and cross-network routing. The gateway VTAM must be either an MVS host running VTAM V2R2 or subsequent releases or a VM host running VTAM V3R1.1 or subsequent releases.

Gateway NCPs

You must specify at least one gateway NCP for each network gateway. The gateway NCP works with the gateway VTAM to control address translation for cross-network sessions. The gateway NCP provides network address translation from a pool of network alias addresses for each network. The gateway NCP also assists in activating explicit and virtual routes for cross-network sessions.

The gateway NCP is a resource in each network to which it connects. Each gateway NCP has one native network, where the gateway VTAM that can activate the NCP's subordinate resources resides. Each gateway NCP also has one or more non-native networks. For example, in Figure 110 on page 439, NCP21 is subarea 21 in NETA and subarea 51 in NETX. NCP21's native network is NETA; its non-native network is NETX.

The configuration in Figure 102 on page 390 is the same as the configuration in Figure 110 on page 439, except that Figure 110 on page 439 shows a multiple-network environment because it contains two independent SNA networks. Note that NCP21 in NETA is also subarea 51 in NETX and NCP11 in NETB is also subarea 52 in NETX. Also note that there are two NCP11s in this environment. The names can be the same because they reside in different networks (hosts, however, cannot have the same name due to CDRM conflicts).

Code everything that you would for Figure 102 on page 390. Add at least the following to the definition of HOST1 in NETA:

- NETID must be coded in the start options.

Add at least the following to the definition of HOST2 in NETA:

- GWSSCP=YES must be defaulted or coded and NETID should be coded in the start options.
- One NETWORK definition statement for each CDRM or group of CDRMs, identifying which CDRMs reside in each network.
- CDRM definition statement in the cross-domain resource manager major node for VTAM3 in NETB.
- A GWPATH definition statement following the cross-domain resource manager minor node for VTAM3 in NETB. Code the ADJNET, ADJNETEL, ADJSA, and GWN or SA operands.
- NETID and GWCTL operands on the PCCU definition statement defining NCP21 in NETA.
- NETID, HSBPOOL, and CANETID operands on the BUILD definition statement in the NCP major node for NCP21 in NETA.

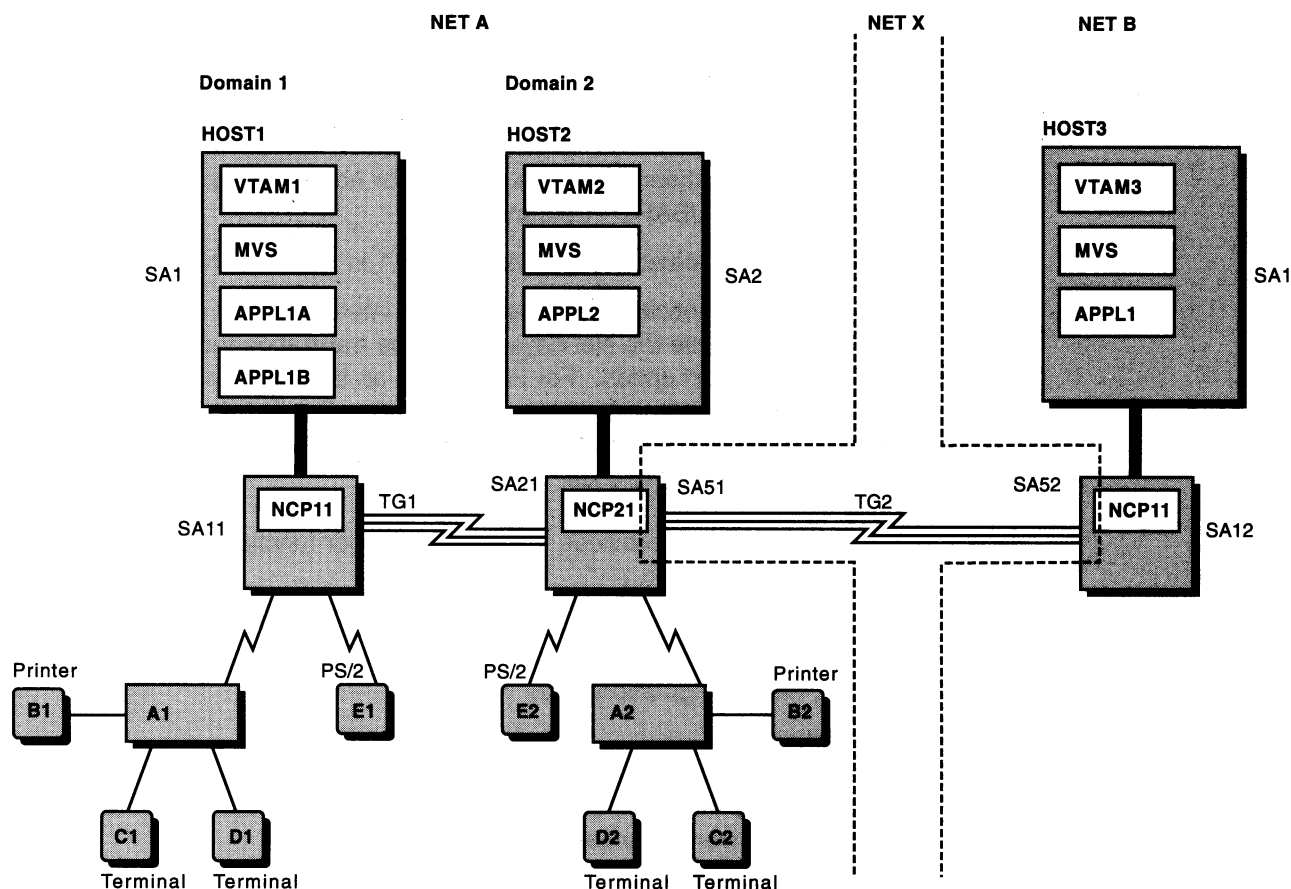


Figure 110. Multiple-Network Environment

- GWNAU definition statement in the NCP major node to identify subarea 52 in NETX.
- NETWORK definition statement in the NCP major node to identify NETX. Code the NUMHSAS and SUBAREA operands.
- GWNAU definition statement to represent NCP21 in NETA as NCP51 in NETX.

Also code at least the following in HOST3 in NETB:

- Start options for HOST3 in NETB (NETID, SSCPID, and SSCPNAME). Code or default to GWSSCP=YES.
- Application program minor node for APPL1 in NETB.
- One NCP major node for NCP11 in NETB.
- One NETWORK definition statement for each CDRM or group of CDRMs, identifying which CDRMs reside in each network.
- CDRM definition statements in the cross-domain resource manager major node for VTAM3 in NETB and VTAM2 in NETA.
- A GWPATH definition statement following the cross-domain resource manager minor node for VTAM2 in NETA. Code the ADJNET, ADJNETEL, ADJSA, and GWN or SA operands.
- NETID and GWCTL operands on the PCCU definition statement defining NCP11 in NETB.

- NETID, HSBPOOL, and CANETID operands on the BUILD statement in the NCP major node for NCP11 in NETB.
- GWNAU definition statement in the NCP major node to identify NCP51 in NETX.
- NETWORK definition statement in the NCP major node to identify NETX. Code the NUMHSAS and SUBAREA operands.
- GWNAU definition statement to represent NCP11 in NETB as NCP52 in NETX.

Note: Resources with identical names can exist in interconnected networks. To resolve naming conflicts, use the NetView alias name translation facility or the NQNMOME=NQNAME start option. For information on the NetView alias name translation facility, see “NetView Alias Name Translation Facility” on page 483. For information on the NQNMOME=NQNAME start option, see “NQNMOME” in the *VTAM Resource Definition Reference*.

SNA network interconnection provides the following major functions:

Cross-network session control

Enables VTAM to perform cross-network session initiation and termination. As required, VTAM can request address translation, name translation, and configuration data.

Address translation

Enables networks with different addressing schemes to communicate. The gateway NCP performs the function of address translation when crossing network boundaries.

Resource name translation

Enables networks with conflicting resource names to communicate. This function can be performed by the NetView program or the VTAM session management exit routine.

Note: The NQNMOME=NQNAME start option is an alternative to resource name translation. For more information, see “NQNMOME” in the *VTAM Resource Definition Reference*.

Cross-network problem determination and network management

Collects and correlates problem determination and network configuration information for cross-network sessions. The NetView program provides the function of cross-network problem determination and network management.

SNI Configurations

This section describes some basic SNI configurations and their attributes.

Single gateway configuration

This configuration consists of one gateway VTAM and one gateway NCP. This configuration is the easiest to configure and define, and cross-network session control is centralized. However, failure of the gateway NCP disrupts all cross-network sessions. This type of gateway can be used to connect up to 255 networks, with the gateway NCP residing in multiple non-native networks.

Single gateway VTAM with multiple gateway NCPs

In this configuration two or more NCPs act as the gateway NCPs. This configuration has the advantages of the single gateway configuration,

but the failure of one gateway NCP does not disrupt all cross-network sessions.

Serial configuration

In this configuration, one gateway VTAM resides in a transport network and controls two or more gateway NCPs that connect to the non-native networks. This configuration is best if the networks on either side of the transport network do not exchange much information.

Back-to-back configuration

This configuration consists of two gateway VTAMs and two gateway NCPs that connect through a null network. This configuration provides the most network isolation and security, but it requires more network definition. It also means that address and name translation must be performed twice, and the cross-network session path is longer. The SSCP-SSCP session between HOST2 and HOST3 in Figure 110 on page 439 is an example of a back-to-back configuration.

Starting VTAM

The start options that you set for VTAM can determine whether this VTAM is a gateway VTAM or a non-gateway VTAM.

Note: You can use the XNETALS start option to connect to non-native type 2.1 PUs. For more information see “Non-Native Network Type 2.1 Connections” on page 250.

Start Options for Gateway VTAMs

To create a gateway VTAM, specify the following start options:

NETID (required)

NETID specifies the name of the network that contains this gateway VTAM. The network identifier allows VTAM to determine which gateway NCP definition statements apply to the gateway host.

SSCPID (required)

Change SSCPID as necessary. SSCPID uniqueness must be maintained across connecting networks because two SSCPs with the same ID cannot establish a session.

SSCPNAME (required)

SSCPNAME specifies the unique name by which this gateway SSCP is known to the gateway NCP and other networks. This name must be the same as the NAME operand on a GWNAU definition statement that is used to predefine an SSCP element address in a gateway NCP. This name must also be the same as the owner name (CDRM operand) used on the CDRSC definition statements in other networks that represent resources owned by this VTAM.

HOSTPU (recommended if the NetView program is installed)

HOSTPU specifies a unique network name for the VTAM host physical unit. The NetView program uses this name to determine which VTAM host physical unit it is tracing. (If HOSTPU is not specified, all VTAM host physical units are named ISTEPUS.)

GWSSCP=YES (required)

GWSSCP=YES indicates that this SSCP can serve as a gateway SSCP.

Note: Even if this SSCP is not performing a gateway function, GWSSCP can be used to enable this SSCP to act as an intermediate SSCP during a session setup.

GWSSCP Start Option for Non-Gateway VTAMs

If you code GWSSCP=NO, trial-and-error routing of session initiation requests in the originating domain can be done, but no rerouting is possible for session initiation requests received by this host from other SSCPs.

Coding GWSSCP=YES for a VTAM that does not activate gateway NCPs permits VTAM to reroute received session establishment and directory requests. It also enables you to code GWPATH definition statements in conjunction with a CDRM definition to define multiple gateway paths to that CDRM.

Configuration Lists for Gateway VTAMs

You might want to update the initial configuration list to include any new major nodes and tables, such as the following:

- Gateway NCP major nodes
- CDRM major nodes
- CDRSC major nodes
- Adjacent SSCP tables
- Path tables.

Application Programs

For a multiple-network environment, you need to define the NetView program or another name translation facility and control application program security across networks.

Note: An application program may become a shadow resource if a CDRSC with the same name already exists when the major node containing the application program's definition is activated. For more information on shadow resources, see "Shadow Resources" on page 407 .

Defining the NetView Program

If you are running the NetView program, ensure that at least one application program major node contains definitions for the NetView program and the NetView alias name translation facility if you are using it. If you are using network-qualified names, it is not necessary to use the NetView alias name translation facility. Network-qualified names provides an alternative to using the facility for resolving duplicate names. The NetView alias name translation facility can be used in conjunction with network-qualified names.

If you want to use the alias name translation facility provided by the NetView program, by the NCCF licensed program, or by a similar user-written application program, include an APPL definition statement for that application program in the application program major node. The name on the APPL definition statement must be ALIASAPL, as shown in the following example:

ALIASAPL APPL AUTH=(CNM),PRTCT=password

Note: You can use a different name than ALIASAPL, but if you do, change and reassemble the CNM routing table. If you want to do this, see “CNM Routing Table (MVS)” in *VTAM Customization* for more information.

The AUTH=CNM operand on the APPL definition statement means that this application program can use the communication network management (CNM) interface. This operand must be coded for all CNM application programs, including the following:

- The session monitor component of the NetView program
- The NetView program alias name translation facility
- The NLDM licensed program
- A user-written application program used for name translation.

The following NetView program definition statements must also be coded:

- One ORIGNET definition statement for each network that requires alias name translation. The definition statements that follow this ORIGNET definition statement apply to the named network.
- One LU definition statement for each LU that is given an alias name.
- Add one COS definition statement for each class-of-service name that must be mapped to the name of an equivalent class of service used in another network.
- One MODE definition statement for each logon mode name that must be mapped to the name of an equivalent logon mode used in another network.

Connecting Networks

When you are planning to connect networks, you should consider the following, and if necessary, discuss these things with the system programmers for the networks that are being connected:

- To guarantee uniqueness, know the SSCPIIDs and the NetView domain IDs used in each network, or use a naming convention.
- If you own a gateway NCP:
 - Obtain the MAXSUBA specification used for any attaching network that contains non-extended subarea addressing nodes so that you can code the MAXSUBA operand on the NETWORK definition statement in the gateway NCP.
 - Specify an available subarea number in the attaching network on the SUBAREA operand on the NETWORK definition statement.
 - Know how many subarea nodes can communicate with the gateway NCP in the attaching network so that you can code the NUMHSAS operand on the NETWORK definition statement in the gateway NCP.
 - Exchange information about explicit route and virtual route structures so that you can code PATH definition statements for non-native networks in the gateway NCP.
- If you plan to code GWPATH definition statements, you need to know the subarea and element addresses of all CDRMs in adjacent networks with which your gateway VTAM is to establish sessions.

- If you are not using network-qualified names, to avoid future name conflicts, decide upon a naming convention for alias names and for real names.
- If you want to guarantee that certain application programs can always obtain an alias address, you need to know the real logical unit names of those application programs. These names are required on the GWNAU definition statements that predefine specific resources in the gateway NCP.
- Tell system programmers for the other networks the names to be used for logical units. Discuss equivalent COS and logon mode entry names.
- Some application programs require a definition for each of their session partners. If these application programs are to have cross-network sessions, define one of the following to the application programs:
 - Alias names for session partners in other networks
 - LU alias names (on the CDRSC definition statements) for session partners in other networks
 - USERVARs that map to the network-qualified names of session partners in other networks.

- If you code a CDRM operand on a CDRSC definition statement and you specify a NETWORK definition statement for it, use the real SSCP name for the host in the other network.

The real SSCP name is the name specified in the SSCPNAME start option. For any host for which SSCPNAME is not specified, the real SSCP name is the name of the host CDRM for that host. You should use this name on your definition for this CDRM if you code such a definition.

- If you code a CDRM definition statement in an adjacent SSCP table, specify the real SSCP name on the label of the CDRM definition statement.
- If you code an SSCP name on an LU definition statement for the NetView alias name translation facility, you must specify the real SSCP name.
- You must code the real SSCP name on the ADJCDRM operand in an adjacent SSCP table.
- Ensure that the COS table named in the COSTAB operand of the BUILD definition statement or a NETWORK definition statement in the gateway NCP is available to the gateway SSCP. You must install it in VTAMLIB. The table specified in the COSTAB operand is used for mapping class-of-service table entries for the routes that originate in the gateway NCP and terminate in the network being defined. If a COS table for the connected network is not supplied, the COS table for your network must include any COS names that can be requested by logical units in other networks.
- Ensure that NCP data transfer related operands (such as UNITSZ, TRANSFR, and BFRS) that are coordinated between domains in a multiple-domain network are also coordinated between interconnected networks.
- If you use the NetView program, you need to know the NetView suppression characters used in every domain in every network to ensure that the characters are all the same.

Defining a Gateway VTAM

The gateway VTAM, in conjunction with the gateway NCP, performs cross-network session initiation and termination by acting as an intermediary between VTAMs in the connected networks. (After the session is set up, there is no need for the LU-LU session path to go through the gateway VTAM.) The gateway VTAM also directs the gateway NCP to set up the address transformations and routes needed for cross-network sessions. The gateway VTAM determines the routing for cross-network sessions and resolves resource names in conjunction with the name translation facility.

Note: A gateway VTAM must be either an MVS host with VTAM V2R2 or later, or a VM host with VTAM V3R1.1 or later.

The gateway VTAM can be in any one of the connected networks.

To convert a non-SNI VTAM to a gateway VTAM, you need to make the following changes:

- Update your start options.
- Update your initial configuration list.
- Update your application program major node if you are using NetView alias name translation facility.
- Update your CDRM major node.
- Create CDRSC definition statements (optional).
- Create an adjacent SSCP table.

In a gateway configuration, VTAM has several different names. The following three names are assigned using start options:

- The ID specified on the SSCPID start option
- The name specified on the SSCPNAME start option
- The name specified on the HOSTPU start option.

These are the names by which this SSCP's CDRM is known by other domains in this and other networks. No requirement exists for these names to match any of the other names listed previously. However, you might want to have each CDRM known by a single name throughout all the networks. You can do this by coding a single set of CDRM definitions that is used in all the connected networks. Coding the same name for this SSCP on all definitions makes network management easier.

Another name can be assigned on the CDRM definition statement. However, if that name differs from the SSCPNAME that is used, the results are unpredictable because of VTAM's unique use of each of the varying names. It is recommended that these names be the same to avoid inconsistencies. One exception to this is that HOSTPU must be different from the CDRM name or SSCPNAME.

Defining a Gateway NCP

The gateway NCP assists in cross-network session initiation and termination by providing network address transformations from the pool of network alias addresses for each network. In addition, the gateway NCP assists in activating explicit and virtual routes for use by cross-network sessions. The gateway NCP is a part of all the networks it connects. The resources attached to the gateway NCP, however, are owned by VTAMs in only one of the attaching networks.

The gateway NCP is attached to other subareas in the various networks. It connects with SDLC links, token rings, or channel attachments.

Every gateway NCP has one native network. The native network is identified by the NETID operand on the BUILD definition statement in the NCP generation definition. The native network contains the VTAMs that are allowed to own (activate) the subordinate resources (the physical units, LUs, peripheral links, and cross-subarea links) of the gateway NCP. The gateway NCP resources can be owned by any or all of the VTAMs in the native network, whether they are gateway VTAMs or not.

Every gateway NCP has one or more non-native networks. Each non-native network is identified by a NETWORK definition statement in the NCP generation definition. A non-native network is one that is attached to the gateway NCP for the purpose of interconnection. A VTAM in a non-native network can activate the gateway NCP but cannot activate any of the NCP's subordinate resources.

At least one gateway NCP is required for network interconnection. The gateway NCP performs the address translation necessary for cross-network session traffic.

A pool of alias addresses is used for cross-network sessions. The coding of the GWNAU definition statement determines whether VTAM can activate SSCP-SSCP sessions in some configurations, because it defines representations of cross-network logical units and SSCPs within the gateway NCP.

An SSCP can neither send nor receive a request to establish a cross-network session with another SSCP unless it knows that SSCP's subarea and element address. A non-gateway SSCP cannot get the address from the gateway NCP; thus, the non-gateway SSCP depends on the GWNAU definition statement corresponding to the other-network SSCP to supply the alias address. You can code this address on the SUBAREA and ELEMENT operands of the CDRM definition statement that defines the other-network SSCP to the non-gateway host.

In a back-to-back gateway configuration, a gateway SSCP must be able to tell its gateway NCP what address represents the other-network SSCP in the second gateway NCP. The address in the GWNAU definition statement for the other-network SSCP in the second gateway NCP is the same as that in the ADJNETSA and ADJNETEL operands of the GWPATH definition statement following the CDRM definition statement that defines the other-network SSCP to the gateway host.

Sometimes a request arrives at a gateway NCP before VTAM has determined the address. In this case, the gateway NCP rejects the request. At the same time, the NCP signals the destination SSCP to set up the session. The NCP knows the address of the destination SSCP only if the SSCP name on the GWNAU definition statement for the alias address to which the ACTCDRM request is sent matches an

SSCP name that the NCP received in an ACTPU request. This SSCP name sent in an ACTPU request from a gateway SSCP is determined by the SSCPNAME start option.

If you are dividing a network into multiple networks and some subarea nodes are not extended network addressing nodes, then group those subarea nodes, if possible, into one or more unique networks. You might want to change MAXSUBA to meet a particular addressing requirement for a newly created network.

If a non-extended addressing node VTAM activates a gateway NCP, the NCP must define the MAXSUBA operand on the BUILD and NETWORK definition statements. A non-extended addressing NCP, functioning as a gateway, does not support sessions with resources in nodes having subareas greater than MAXSUBA. The non-extended addressing node gateway cannot activate an explicit route to a subarea greater than MAXSUBA in that network.

Converting a Non-SNI NCP to a Gateway NCP

To define a gateway NCP, code a NETID and a GWCTL operand on the PCCU definition statement in your NCP definition.

NETID specifies the network containing the host node that the PCCU definition statement defines. The NETID and SUBAREA operands are used together to determine which VTAM host should process this PCCU definition statement. If NETID is omitted, only the SUBAREA value is used. This could result in the PCCU definition statement being processed by the wrong VTAM host.

The GWCTL operand tells VTAM whether other gateway SSCPs can perform some of the cross-network session setup functions. GWCTL specifies this SSCP's role only for the gateway NCP being defined. It applies only if this SSCP is on the session setup path.

In addition to defining the gateway NCP itself, define the network that is native to it and the non-native networks.

Defining Native Networks

The native network contains the VTAM that is allowed to own and activate this gateway NCP. The native network is identified by the NETID operand on the BUILD definition statement.

Coding the BUILD Definition Statement

Identify the NCP's native network with the NETID operand on the BUILD definition statement.

On the BUILD definition statement, add the following operands:

NETID (required)

The NETID operand specifies the name of the native network. VTAM compares the NETID coded on the BUILD definition statement to the NETID start option coded for VTAM. If these values match, VTAM is in the NCP's native network. Resources attached to an NCP are controlled only by SSCPs in the native network of the NCP.

If the NETID specifications do not match, VTAM searches for a NETWORK definition statement with a matching NETID.

HSBPOOL (required)

Two half-session control blocks (HSCBs) are required to represent each cross-network session, one for each session partner. HSCBs can be statically allocated with the GWNAU definition statement.

The value specified for the HSBPOOL operand is the total number of HSCBs in the gateway NCP to represent cross-network sessions that require dynamic assignment of an HSCB. HSCBs for each network attached to the gateway NCP are dynamically allocated from this pool. You can limit the number of HSCBs that can be allocated from the pool for a given network by specifying a NETLIM value on the BUILD or NETWORK definition statement. You can also limit the HSCBs for a given address by specifying SESSLIM on the BUILD or NETWORK definition statement.

CANETID

The CANETID operand applies only to a channel-attached gateway NCP and the 3720 communication controller. CANETID specifies the network ID of the network containing the host attached through each channel adapter. The gateway NCP uses the CANETID specification to determine the network affinity for each specified channel adapter. CANETID thus specifies in which network each channel-attached host is located.

The gateway NCP uses CANETID to determine which channel-attached hosts can activate resources attached to the gateway NCP. It also determines whether a particular channel-attached host is in the native network.

Also do the following:

- Add one ERLIMIT operand (optional).
- Add one SALIMIT operand (optional).
- Add one GWSESSAC operand.
- Add one GWAEXIT operand (optional).
- Add one COSTAB operand (optional VTAM-only operand).
- Update the VRPOOL to include the additional virtual routes from other networks that end in the gateway NCP.

Coding the GWNAU Definition Statement

For the native network, place a GWNAU definition statement after the SYSCNTRL definition statement:

```
BUILD NETID=neta
SYSCNTRL
GWNAU NAME=xname,NETID=yname,ELEMENT=nn
```

NAME identifies a CDRM for the connected network

When you code the CDRM definition for this resource in your gateway VTAM, this name must match the CDRM name.

NETID identifies the network in which this CDRM resides

This value must match the value for the NETID operand of the BUILD definition statement found in the other network's gateway NCP.

ELEMENT reserves an element address in the native network to represent the non-native CDRM.

This value must match the ELEMENT value in the GWPATH statement for gateway VTAM.

GWNAU reserves one or more element addresses in the native network to represent other network resources. Some elements can be reserved in a dynamic pool while others can be predefined (that is, associated with a particular resource name).

Following are two circumstances when the GWNAU definition statement must be coded to predefine an element address in the gateway NCP:

- If a non-gateway SSCP in the network is to establish an SSCP-SSCP session with a gateway SSCP in another network, the GWNAU definition statement must be coded to predefine an element address for a gateway SSCP. This element address must also be specified on the ELEMENT operand of the CDRM definition statement that defines the gateway SSCP to the non-gateway SSCP.
- If you have cross-network sessions between CDRMs in non-adjacent networks, the GWNAU definition statement must be coded to reserve an element address in the gateway NCP subarea within the adjacent network. This element address must be specified on the ADJNETEL operand of the GWPATH definition statement that defines the non-adjacent CDRM.

Optionally, application programs that have multiple cross-network sessions can be predefined so that half-session control blocks (HSCBs) required to support the sessions with the application program are preassigned to the element used to represent the application program in the gateway NCP. This ensures that at least the preassigned number of control blocks are available to represent sessions with the application program. Statically allocated HSCBs and addresses cannot be used by any other resource.

Note: If you plan to have session monitors communicating from one network to another, it is strongly recommended that you code two GWNAU definition statements for each network management session (one for the primary-to-secondary session and one for the secondary-to-primary session) to reserve session control blocks in the NCP for the network management session.

Coding the HOST Definition Statement

On each HOST definition statement defining VTAM, you should add one NETID operand. This is not required but is recommended. If NETID is omitted, only the SUBAREA value is used. This can result in the HOST definition statement being processed by the wrong VTAM host.

NETID specifies the network containing the host node to which this HOST definition statement applies. The NETID and SUBAREA operands are used together to determine which VTAM host should process this HOST definition statement.

Update NCP PATH definition statements by doing the following:

- Define the new routes that originate in the gateway NCP subarea.
- Map virtual routes to explicit routes for routes that originate in the gateway NCP subarea. For virtual routes that terminate in a VTAM host subarea, the forward and reverse virtual route numbers must match.

On each PU definition statement that represents a type 4 physical unit, add one NETID operand (required).

NETID specifies the NETID of the network in which the physical unit, represented by this link station, resides.

Defining Non-Native Networks

Use the NETWORK definition statement in the NCP major node to identify each network attached to a gateway NCP.

To define the non-native networks, the following changes are required:

1. Add one or more NETWORK definition statements (one is required for each non-native network).

NETWORK identifies a non-native network that attaches to the gateway NCP. On the NETWORK definition statement, the following operands can be coded:

- Add one ACTPU operand (optional).

ACTPU specifies whether SSCPs in the non-native network can activate the gateway NCP.

- Add one COSTAB operand (optional VTAM-only operand).

COSTAB specifies a user-written class-of-service table to be used for routes that originate in the gateway NCP and that are within the non-native network being defined. No table is needed if the only COS table entries used are ISTVTCOS and the default class of service.

- Add one MAXSUBA operand (optional).

This operand specifies the MAXSUBA value for the non-native network. It is not required if all nodes in that network use extended network addressing.

- Add one SALIMIT operand (optional).

The SALIMIT operand specifies the maximum number of subareas that can be used in the non-native network.

- Add one NUMHSAS operand (required).

This operand specifies the number of subarea nodes in the non-native network that can communicate concurrently with the gateway NCP.

- Add one SUBAREA operand (required).

This operand specifies the subarea address of the gateway NCP as known in the non-native network.

2. Add one or more GWNAT definition statements (required).

Add one or more GWNAT definition statements after the NETWORK definition statement when defining non-native networks.

```
NETWORK NETID=netb
GWNAU NETID=neta,NAME=cdmname,ELEMENT=nn
```

NETID identifies the network in which the following CDRM name resides to the non-native network

This value must match the value for the NETID operand in the BUILD definition statement.

NAME identifies a CDRM to the non-native network

ELEMENT reserves an element address in the non-native network to represent that CDRM.

Some elements can be reserved in a dynamic pool, while others can be predefined. A predefined element is one that is associated with a particular resource name. Each resource that has a cross-network session with a resource in this network must have at least one address. (Some application programs require more than one address if they have multiple sessions.)

Note: If you plan to have session monitors communicating from one network to another, it is strongly recommended that you code two GWNAU definition statements for each network management session (one for the primary-to-secondary session and one for the secondary-to-primary session) to reserve session control blocks in the NCP for the network management session.

Optionally, you can preassign session related control blocks (HSCBs) to application programs (or SSCPs) that have many cross-network sessions. The HSCBs are preassigned to the element that represents the application program in the gateway NCP. This ensures that at least the preassigned number of control blocks are available to represent sessions with the application program.

3. Update NCP PATH definition statements by:

- Adding new NCP PATH definition statements to define routes that originate in the gateway NCP subarea for the network identified by the previous NETWORK definition statement. See “Cross-Network Routing” on page 473.
- Mapping virtual routes to explicit routes for these routes. For virtual routes that terminate in a VTAM host subarea, the forward and reverse virtual route numbers must match. See “Cross-Network Routing” on page 473.

Dynamic Network Identification

A single gateway NCP can backup any number of similarly configured gateway NCPs. A gateway VTAM accepts the names of the networks connected to a gateway NCP, without requiring any more definition than NCP already requires. Any VTAM connected directly to a gateway NCP through a leased SDLC line can transmit its own NETID.

Dynamic network identifier enables you to dynamically add and delete NETIDs (networks) within a gateway NCP. Its primary purpose is to provide more back-up flexibility; a single gateway NCP can be used to back-up numerous NCPs with different NETIDs (residing in different networks). It might also be useful in situations where a gateway NCP occasionally uses links in other networks, especially on a random basis.

The gateway NCP implementing the dynamic NETID must be an NCP Version 5 Release 4. The gateway VTAM that owns the gateway NCP implementing the

dynamic NETID must be in a host running MVS/ESA or VM/ESA. VTAMs that are connecting the gateway NCP can run VSE/ESA, * MVS/ESA, or VM/ESA.

Enabling Dynamic NETID

To enable dynamic NETID code a model network without the NETID and SUBAREA coded. Do the following:

1. Define NETWORK definition statements for all the networks for which you want continuous contact.
2. Define a model network statement with the configuration of any other network the NCP can contact.
3. Control the number of networks represented by a model with which the NCP can have contact simultaneously by specifying the number in the COPIES operand of the NETWORK definition statement for the model network. The COPIES operand is required. If you specify COPIES=3, three model networks are created in the gateway NCP.

Creating Dynamic Networks from Model Networks

When the gateway NCP receives contact from a network that has an unknown NETID, NCP accepts the NETID in the XID exchange. NCP uses an available pool network (which is a copy of the model network), assigns the unknown NETID to it, and marks it as dynamic, thus dynamically creating the network. As long as there are users of the dynamic network (the INN links are active), the network remains in existence. After there are no more active users of the network, the DYNAMIC network is cleared and returned to the pool, making it available for use by another network.

When a network is dynamically created within a gateway NCP, all gateway VTAMs that have activated the physical unit of that NCP are notified and made aware of the new network. Also, whenever a gateway VTAM activates the physical unit of a gateway NCP, that VTAM is notified of all dynamic and static (predefined) networks. VTAM is also notified when dynamic networks are cleared and no longer exist.

Note: VTAMs prior to Version 3 Release 4 and non-gateway VTAMs are not notified about the addition or deletion of dynamic networks.

Dynamic network capability is supported on leased, subarea dial, and NTRI INN links. The protocols used for dynamic NETID depend on whether the link is a leased (including NTRI) or switched line. For detailed information on the NETID negotiation protocols used for link activation between two NCPs, refer to the *NCP, SSP, and EP Resource Definition Reference*.

Considerations for Implementing Dynamic Network Identifier

You need to consider the following when implementing dynamic network identifier:

- All dynamically created networks have the same subarea, but have different NETIDs.
- The model network must have paths defined to at least the first adjacent subarea that is contacted when the NETID is initially established. After this, you can use dynamic path update to dynamically define paths. See “Dynamic Path Update” on page 313. Paths require careful planning, especially when the paths go to the same subarea number in several different networks.

Note: When a dynamic network is cleared, any paths created by dynamic path update for that network are also cleared and lost.

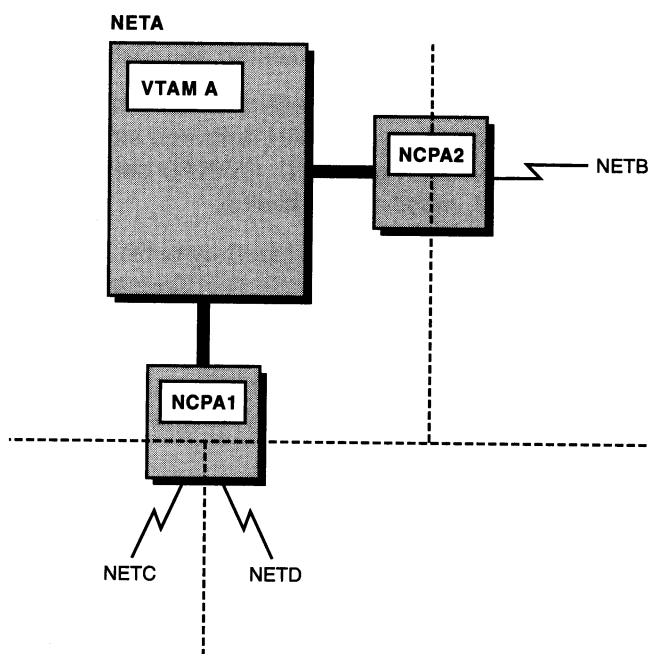
- Because the model network represents other networks, the model network must contain all necessary GWNAU definition statements for any NETID of any network that the model represents. GWNAU definition statements cannot be dynamically added, modified, or deleted.
- For a link station and its line to support dynamic NETID, the GROUP definition statement for the INN link must have NETID=(name,ANY) coded as a parameter. For leased and switched links, this is coded on the respective GROUP definition statements. For NTRI links, this is coded on the LOGICAL GROUP(s). To establish dynamic NETID capability, code NETID=ANY on the GROUP definition statement. All INN links defined under the dynamic GROUP have the ability to use it.

Note: When coding NETID=(name,ANY) on the GROUP, *name* must be either the default or predefined network within the gateway NCP. If *name* is omitted and NETID=(,ANY) is coded, the default network is assumed.

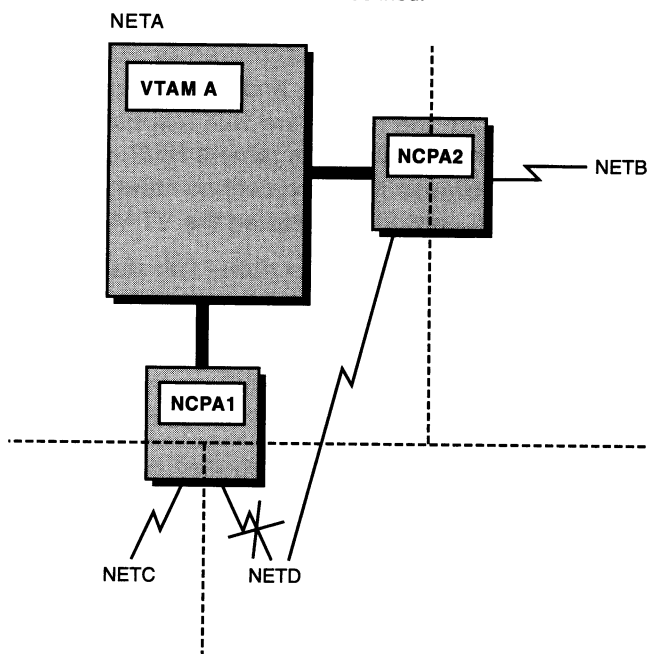
- Ensure that the necessary GWPATH definition statements are coded for the cross-domain resource managers that you are contacting.
- To display static and dynamic networks, use the command
D NET,COS,ID=*nep_name*,NETID=*. This displays the COS table associated with a NETID. The model network is displayed as *NETWORK. Any static networks appear above the model network, and any dynamic networks are displayed below it. Remember, the results are with respect to the VTAM where you issue the command. When displaying a remote NCP, a dynamic network might appear as static because it is a predefined network in the local NCP or because it is in the network that issued the VTAM network identifier.
- To display the network identifier of active link stations, use the command
D NET,STATIONS,A.
- Leased lines can change their network identifier to a predefined network identifier. To do this, code NETID=(name,ANY) on the GROUP definition statement for the leased line. A model network is not needed.
- You need to ensure that the value specified on the HSBPOOL operand on the BUILD definition statement is large enough to accommodate the number of addresses needed to represent all SSCPs and logical units that participate in cross-network sessions (specified on the NUMADDR operand on the GWNAU definition statement) for all predefined and dynamic networks.

Sample Dynamic NETID Configuration

Figure 111 on page 454 is a sample dynamic NETID configuration.



NCPA1 and NCPA2 are gateway NCPs. NCPA1 supports NETC and NETD; NCPA2 supports NETB. NCPA2 has an extra switched line defined.



NCPA1's link to NETD goes down. NETD dials NCPA2 to connect to NETA.

Figure 111. Backup of a Gateway NCP

Following is the coding for the backup gateway NCP (NCPA2):

```

DNETID  BUILD VERSION=V5R4,
:
GLS      GROUP LNCTL=SDLC,ACTIVTO=420.0,TYPE=NCP,DIAL=NO,REPLYTO=3, *
          SDLCST=(STPRIM,STSECD),PUTYPE=4, *
          NETID=(NETA,ANY)
LLS      LINE  ADDRESS=(10,FULL), *
          DUPLEX=FULL, *
:
PLS      PU    PUTYPE=4,TGN=56, *
```

```

      RETRIES=(,2,0)
GSW    GROUP LNCTL=SDLC,ACTIVTO=0,DIAL=YES,REPLYTO=3,PUTYPE=4,      *
      SDLCST=(STPRIM,STSECD),V25BIS=(YES,DLS DLC),                *
      NETID=(NETB,ANY)
LSW    LINE  ADDRESS=(035,FULL),                                     *
:
NETB   NETWORK SUBAREA=9,NUMHSAS=12,NETID=NETB,                     *
      PATHEXT=2,TGBXTRA=3
      GWN AU ELEMENT=6,NETID=NETA,NAME=SSCPA,NUMSESS=4
      GWN AU NUMADDR=7
      PATH  DESTSA=19,                                             *
      ER0=(19,1),ER1=(19,1),ER2=(19,1),ER3=(19,1),               *
      ER4=(19,1),ER5=(19,1),ER6=(19,1),ER7=(19,1),               *
      VR0=1,VR1=1,VR2=2,VR3=3,VR4=4,VR5=5,VR6=6,VR7=7
      PATH  DESTSA=(8,18),                                         *
      ER0=(8,36),ER1=(8,36),ER2=(8,36),ER3=(8,36),               *
      ER4=(8,36),ER5=(8,36),ER6=(8,36),ER7=(8,36),               *
      VR0=1,VR1=1,VR2=2,VR3=3,VR4=4,VR5=5,VR6=6,VR7=7
COPIES NETWORK SUBAREA=5,NUMHSAS=12,COPIES=6,                     *
      PATHEXT=6,TGBXTRA=5
      GWN AU ELEMENT=8,NETID=NETA,NAME=SSCPA,NUMSESS=3
      GWN AU ELEMENT=15,NETID=NETB,NAME=SSCPB
      GWN AU NUMADDR=8
      PATH  DESTSA=15,                                             *
      ER0=(15,3),ER1=(15,3),ER2=(15,3),ER3=(15,3),               *
      ER4=(15,3),ER5=(15,3),ER6=(15,3),ER7=(15,3),               *
      VR0=1,VR1=1,VR2=2,VR3=3,VR4=4,VR5=5,VR6=6,VR7=7
      PATH  DESTSA=24,                                             *
      ER0=(24,56),ER1=(24,56),ER2=(24,56),ER3=(24,56),          *
      ER4=(24,56),ER5=(24,56),ER6=(24,56),ER7=(24,56),          *
      VR0=1,VR1=1,VR2=2,VR3=3,VR4=4,VR5=5,VR6=6,VR7=7
GENEND GENEND
      END

```

Notes:

1. Omit the NETID operand from the NETWORK definition statement.
2. Ensure that you have PATH definition statements defined for all the adjacent subareas. Thus, for the above coding, the adjacent subarea in NETD must be either 15 or 24.

Defining Cross-Domain Resource Managers

You need to code CDRM definition statements for VTAMs in other networks with which this VTAM requires direct communication. In a CDRM major node:

1. Add one NETWORK definition statement for each network to be connected through a gateway NCP. This is necessary to identify the networks that contain the defined CDRMs.

The NETID operand of the NETWORK definition statement identifies the network that contains the defined CDRMs. All the CDRMs that follow a given NETWORK definition statement must belong to that network.

The value coded for NETID on the NETWORK definition statement must match the NETID start option specified for the CDRM in its own domain. CDRM definition statements for CDRMs in the same network as this host CDRM can

follow a NETWORK definition statement whose NETID identifies the network that contains this host CDRM.

By coding a NETWORK definition statement for every network (including that of the host CDRM), you can use the same set of CDRM definitions at hosts in different networks. (However, these hosts must all have VTAM V2R2 or later releases.)

2. Add one CDRM definition statement for each SSCP in another network with which the gateway VTAM is in session. (This is required.)

The CDRM names defined in the gateway SSCP must be unique and must immediately follow their own NETWORK definition statement.

3. Add one or more GWPATH definition statements after each CDRM definition statement that defines an SSCP in another network. If you specify the GWSSCP=YES start option, you can use a GWPATH definition statement in conjunction with a CDRM definition statement. In this case, the value of the ELEMENT operand on the GWPATH definition statement must match that defined with a GWNAU definition statement in the gateway NCP through which the gateway SSCP is reached. If the gateway SSCP can be reached through multiple gateway NCPs, then multiple GWPATH definition statements can be associated with the single CDRM definition statement defining the gateway SSCP. (This is required for CDRMs in other networks that are not predefined in the gateway NCP.)

GWPATH definition statements define one or more cross-network session paths to be used for cross-network LU-LU and SSCP-SSCP sessions.

If no GWPATH definition statement is coded, VTAM uses the SUBAREA value specified on the CDRM definition statement to select a gateway NCP. The ELEMENT value on the CDRM definition statement must match that specified by the GWNAU definition statement for a predefined CDRM in the gateway NCP.

The ADJNETSA and ADJNETEL operands on the GWPATH definition statement must specify the subarea and element numbers of the CDRM as they are known in the adjacent network. For a back-to-back gateway NCP configuration, these numbers are obtained from the SUBAREA and ELEMENT operands on the GWNAU definition statement that defines this external CDRM in the second gateway NCP in the back-to-back configuration. For information on how the gateway path is selected, see "Gateway Path Selection" on page 475.

The ADJNETCS operand on the GWPATH definition statement allows you to specify the class-of-service name to be used for establishing the route for the SSCP-SSCP session. Usually, this class-of-service name is ISTVTCOS. If you specify ADJNETCS, VTAM does not invoke the NetView alias name translation facility or the session management exit routine to perform class-of-service translation.

4. The value you code on the SUBAREA operand of the CDRM definition statement should be the subarea in the gateway NCP through which the gateway SSCP is reached. The ELEMENT operand for the CDRM definition statement must match the element predefined for the gateway VTAM with a GWNAU definition statement in that gateway NCP. If these operands are not defined, it is impossible to initiate an SSCP-SSCP session between the non-gateway VTAM and a gateway SSCP in another network.

If the gateway SSCP can be reached through multiple gateway NCPs, it appears as multiple CDRMs (that is, as SSCPs in different subareas) from the point of view of the non-gateway SSCP. If a gateway NCP failure disrupts the SSCP-SSCP session between the non-gateway SSCP and the gateway SSCP, reactivation of the session through a different gateway NCP is possible if you provide multiple CDRM definition statements for the gateway SSCP. Each of these definition statements must have a unique label. The SUBAREA operand specifies the subarea of one of the gateway NCPs, and the element value is reserved for the gateway SSCP with a GWNAU definition statement coded in that gateway NCP subarea. Given these definitions, the operator at either the gateway SSCP or the non-gateway SSCP can re-establish the SSCP-SSCP session through an alternate gateway NCP.

Figure 112 is an example of an SNI back-to-back configuration.

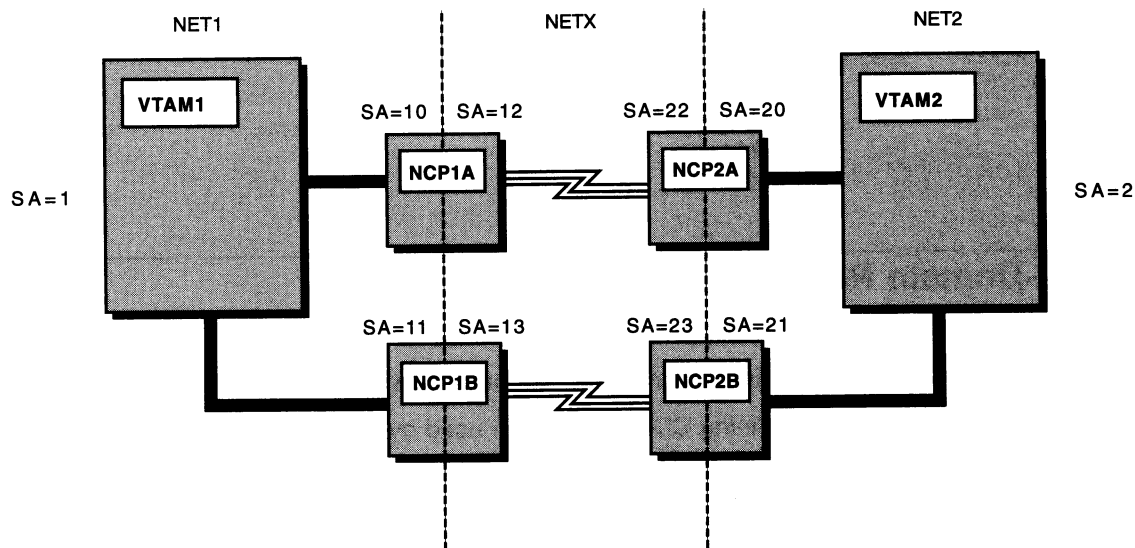


Figure 112. Multiple-Network Configuration: CDRM Major Nodes

Code the following in the CDRM major node in VTAM1:

```

NET1      NETWORK NETID=NET1
VTAM1     CDRM  SUBAREA=1,
:
NET2      NETWORK NETID=NET2
VTAM2     CDRM
NCP1A     GWPATH SUBAREA=10,
           ELEMENT=1,
           ADJNET=NETX,
           ADJNETSA=22,
           ADJNETEL=1,
NCP1B     GWPATH SUBAREA=11,
           ELEMENT=1,
           ADJNET=NETX,
           ADJNETSA=23,
           ADJNETEL=1,

```

It is recommended that you code the ELEMENT operand rather than accepting the default. This way, it is easier to ensure that you do not have the same subarea and element number for multiple resources.

The CDRM major node in VTAM1 shows that you need to code a GWPATH definition statement for each gateway NCP. Note that you need to be careful to match the NCP pairs (NCP1A with NCP2A and NCP1B with NCP2B) to ensure that the CDRM-CDRM session uses the same pair of gateway NCPs.

Thus, code the following in the CDRM major node in VTAM2:

```

NET2      NETWORK NETID=NET2
VTAM2     CDRM  SUBAREA=2,
:
NET1      NETWORK NETID=NET1
VTAM1     CDRM
NCP2A     GWPATH SUBAREA=20,
           ELEMENT=1,
           ADJNET=NETX,
           ADJNETSA=12,
           ADJNETEL=1,
NCP2B     GWPATH SUBAREA=21,
           ELEMENT=1,
           ADJNET=NETX,
           ADJNETSA=13,
           ADJNETEL=1,

```

Defining Cross-Domain Resources

Following are methods of defining resources in other networks:

- Use a CDRSC definition statement without an owning network specified.

This method of defining CDRSCs can be used by hosts within either of the networks in which the session partners reside or by intermediate networks. The definition can be used to satisfy either a real or an alias name, but if the NetView alias name translation facility or the alias function in the session management exit is used, you cannot define a CDRSC for both. For example, in HOST1 (in Figure 114 on page 462), you can code the following:

```

CDRSCSEG  VBUILD      TYPE=CDRSC,...
TERM1     CDRSC       CDRM=HOST2,...

```

To code the NQNMODE or LUALIAS operand for a CDRSC, the owning network must be specified.

- Use a CDRSC definition statement with an owning network specified.

This method of defining CDRSCs can be used by any of the hosts involved in establishing the requested session. This method provides for the best session setup performance. For example, in HOST1 (in Figure 114 on page 462), you can code the following:

```

CDRSCSEG  VBUILD      TYPE=CDRSC,...
           NETWORK     NETID=NETB
TERM1     CDRSC       CDRM=HOST2,...

```

This is the method to use if you want to define cross-domain resources by their network-qualified names. To define cross-network resources by only their network-qualified names, specify NQNMODE=NQNAME (in the definition or using the start option). To define cross-network resources by both their network-qualified names and their unqualified names, specify NQNMODE=NAME (in the definition or using the start option).

The LUALIAS operand is valid only for cross-network resources with an owning network specified.

- Use a dynamically defined CDRSC.

Instead of predefining CDRSCs, you can decide whether some or all CDRSCs are dynamically allocated and defined. The host that owns the logical unit initiating a session can dynamically define the requested session partner. In the intermediate SSCP, both the originating LU and the destination LU can be dynamically defined. In the destination LU's SSCP, the originating LU can be dynamically defined. Independent LUs acting as either the destination or the originating LU can be dynamically defined at either the destination or originating LU's SSCP.

Dynamically defined CDRSCs take the NQNMODE value that is defined on the NQNMODE start option. The NQNMODE start option affects dynamically defined CDRSCs in the following way:

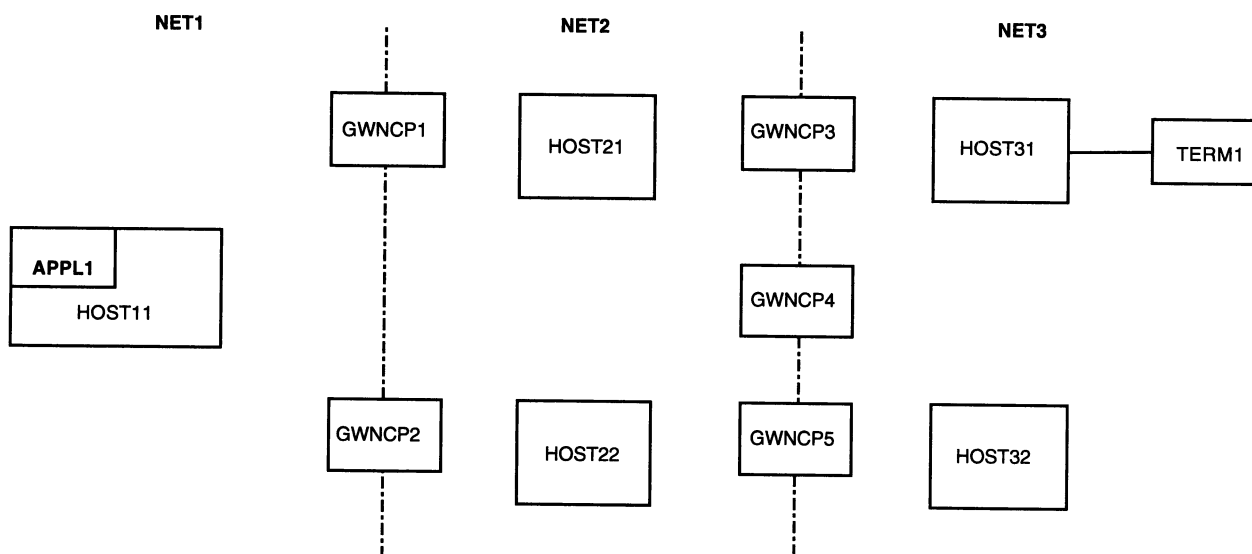
- If NQNMODE=NAME, a cross-network dynamic CDRSC is defined by its network-qualified name and by its alias name and alias network, even if the alias name and alias network are assumed (that is, the alias name is equal to the real name and the alias network is equal to the network of the session partner).
- If NQNMODE=NQNAME, a cross-network dynamic CDRSC is defined by its network-qualified name. A cross-network dynamic CDRSC is not defined by its assumed alias name and alias network unless the name portion of the alias name and the real name are different.

Session Initiation Request

In Figure 113 on page 460, assume that HOST11 and HOST21 are gateway VTAMs, and APPL1 wants to initiate a session with TERM1. The value of NQNMODE, either as a start option or defined on the CDRSC definition statement affects the name by which the cross-network resource is known. If NQNMODE=NAME is specified as a start option or on the CDRSC definition statement for TERM1, because there is no TERM1 in NET1, TERM1's real name is TERM1 in NET3 and its alias name is TERM1 in NET1. (If a resource TERM1 actually existed in NET1, TERM1 in NET3 must be translated to a name that is not being used in NET1, such as TERM3 in NET1.)

If NQNMODE=NQNAME is specified as a start option or on the CDRSC definition statement for TERM1, TERM1's real name is NET3.TERM1. It is **not** known by the alias name NET1.TERM1. (If a resource TERM1 actually existed in NET1, and the session initiation request used a non-network-qualified name, the session would be established with TERM1 in NET1. If the session initiation request used NET3.TERM1, the session would be established with TERM1 in NET3. Another option to using a network-qualified name in a session request would be to translate TERM1 to a name that is not being used in NET1, such as TERM3 in NET1.)

Note: A network-qualified session initiation request (for example, NET3.TERM1) cannot use an application's ACBNAME unless it is the same as its APPL name.



Note: The connections between hosts and NCPs have been omitted for clarity. You can assume that each NCP connects to each host in the networks to the left and right of it.

Figure 113. Example of Three Interconnected Networks

The information in the session initiation request that is flowing from HOST11 to HOST21 is one of the following:

- If APPL1 initiates a session for TERM1 with a non-network-qualified name, and the CDRSC for TERM1 is dynamically defined or predefined using an alias name in HOST11, the session initiation request carries only the alias name (TERM1 in NET1).
- If APPL1 initiates a session for TERM1 using a network-qualified name (NET3.TERM1), or if the session initiation request is network qualified, the session initiation request carries both the alias name (TERM1 in NET1) and the real name (TERM1 in NET3).

The request flow described here assumes that GWCTL=SHR is in the host's PCCU definition statement. If, according to the rules for gateway control described later in this chapter, HOST11 is designated to request an alias address for GWN1, the real name for TERM1 must be properly determined in HOST11 by defining a CDRSC with NETWORK specified or by using the NetView alias name translation facility.

Name Assumption

In Figure 113, assume that HOST21 is a gateway VTAM. Also assume that HOST11 sends only the alias name to HOST21 in the session initiation request. The following steps occur:

1. HOST21 either dynamically defines a CDRSC for TERM1 or uses a predefined CDRSC.
2. HOST21 attempts to determine the real name of TERM1 using the NetView alias name translation facility or the session management exit routine.
3. HOST21 routes the session setup request to HOST31 using the adjacent SSCP tables, and the request leaves the first gateway.
4. When the request leaves the first gateway, VTAM does the following:

- a. If the real name and network identifier are not determined using the NetView alias name translation facility or the VTAM session management exit routine, VTAM sends out a search using the alias name to find the real name and network identifier.
 - b. The alias CDRSC representation (TERM1 in NET1) is updated to reflect the real CDRSC representation (TERM1 in NET3).
5. In SSCP21, a CDRSC is defined or dynamically created for APPL1 also. Because APPL1 is the originating LU, its real name and network identifier are known. The predefined CDRSC can be defined with or without a network identifier.
 6. HOST31 receives the session initiation request from HOST21, which contains both real and alias names for TERM1. HOST31 then determines that it owns TERM1.
 7. HOST31 locates a CDRSC representation for APPL1 or dynamically defines one using the real name (APPL1 in NET1). If you have defined a CDRSC definition statement for APPL1 in NET3, HOST31 will use that CDRSC definition statement and will not use the dynamically defined CDRSC.
 8. When the CDRSC is found, HOST31 merges any pertinent information from the alias CDRSC representation with the real representation.
 9. If session initiation request processing is successful, HOST31 sends a positive response.
 10. HOST11 receives the response, and further processing depends on how TERM1 is defined in HOST11.
 - If TERM1 is dynamically allocated, VTAM merges the acquired information (real network ID and owning SSCP name) into the existing CDRSC.
 - If TERM1 is predefined using its alias name, the predefined CDRSC representation is updated with the real name, network identifier, and owning SSCP.
 - If TERM1 is predefined using its real name and network identifier, no further processing is required. Session setup continues, and resources are recognized by their real names, alias names, and network identifiers.

If both the real and alias names are passed in the session initiation request, either a predefined real CDRSC representation for TERM1 in NET3 is located, or a dynamic CDRSC is allocated. A predefined CDRSC without a network identifier specified can be located. In any case, the resource name is not passed to the alias name translation facility to perform name translation. However, VTAM can invoke the facility to determine if the owning SSCP is defined to the alias name translation facility.

With interconnected networks, defining CDRSCs becomes more complex as cross-network sessions, alias name translation, and adjacent SSCP rerouting are introduced. The configuration in Figure 114 on page 462 shows two interconnected networks.

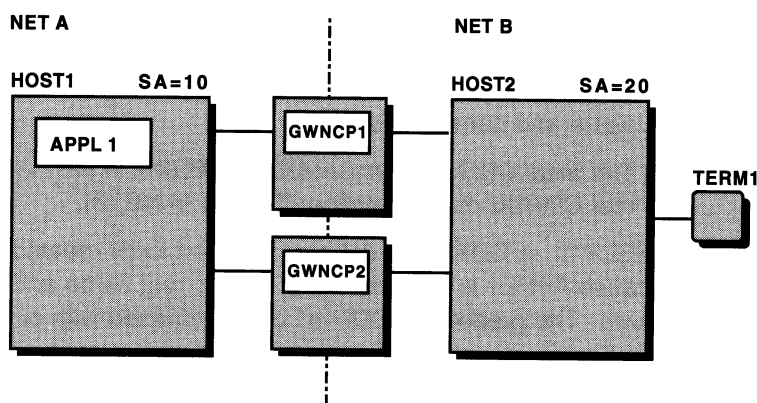


Figure 114. Example of Two Interconnected Networks

To establish cross-network sessions between APPL1 in NETA and TERM1 in NETB, the following CDRM definitions are required if sessions are initiated from either HOST1 or HOST2. This CDRM major node is filed in HOST2 for the external CDRM HOST1.

	NETWORK	NETID=NETA
HOST1	CDRM	CDRDYN=YES,CDRSC=OPT
	GWPATH	GWN=GWNCP1,ADJNETSA=10,ADJNETEL=1,...
	GWPATH	GWN=GWNCP2,ADJNETSA=10,ADJNETEL=1,...

The following CDRM major node is filed in HOST1 for the external CDRM HOST2:

	NETWORK	NETID=NETB
HOST2	CDRM	CDRDYN=YES,CDRSC=OPT
	GWPATH	GWN=GWNCP1,ADJNETSA=20,ADJNETEL=1,...
	GWPATH	GWN=GWNCP2,ADJNETSA=20,ADJNETEL=1,...

Predefined Cross-Domain Resources without Network Specification

If you do not code the NETWORK definition statement and you want to predefine CDRSCs, then you should specify the adjacent SSCP rather than the owning SSCP in the CDRM operand of the CDRSC definition statement.

Note: You cannot specify the NQNMODE or LUALIAS operand for cross-network resources without specifying a NETWORK definition statement.

The name can represent the alias or real name for the resource. When the real name and NETID are determined, VTAM updates the predefined CDRSC to represent the real name.

Note: If the real NETID is cross-network, no other same-network resource can be identified by the originally coded name.

When the last session for that CDRSC ends, VTAM restores the originally coded name so that the CDRSC is the same as that previously predefined. However, if NETID is coded, whether in this network or another, that indicates a real CDRSC.

You can define TERM1 (the destination LU in Figure 113 on page 460) in the following way:

```

      VBUILD TYPE=CDRSC,...
TERM1  CDRSC  CDRM=HOST21,...

```

The CDRM, HOST21, is not the owning SSCP. Rather, it is the SSCP adjacent to the origin logical unit's owning SSCP (HOST11). Because the CDRSC is defined in HOST11's network (or is allowed to default), the resource name (TERM1) is assumed to be the name known in HOST11's network (that is, the alias name).

Following are some advantages of predefining CDRSCs using this method:

- Coding the CDRM operand on the CDRSC definition statement allows you to select the next SSCP in the session setup path. For simple networks that do not plan to provide alternate adjacent SSCPs, this method can eliminate the need for defining adjacent SSCP tables. For more complex configurations, this method of defining CDRSCs identifies the first SSCP that should be tried when alternate adjacent SSCPs are available. In this case, VTAM reorders the adjacent SSCP table you defined; the first SSCP in the table is now the SSCP named on the CDRM operand of the CDRSC definition statement, causing this CDRM to be used first for routing.
- For some configurations, new CDRSC definitions are not required. If a multiple-network configuration is the result of splitting a single network and if the owning CDRMs have not changed, the CDRSC definitions previously defined still cause the request to be routed to the correct SSCP.

The disadvantage of predefining CDRSCs with this method is that the length of the session setup path is increased, though not as much as with dynamically defined CDRSCs using the resource's alias name (because no definitions have to be built dynamically). When the real SSCP name and NETID are determined, VTAM updates the predefined CDRSC to represent the real name. When the last session for that CDRSC ends, VTAM restores the alias name so that the CDRSC is the same as originally predefined and restores the CDRM if it is coded. The NQNMODE and LUALIAS operands are only valid for cross-network CDRSCs with an owning network specified.

This method of predefining CDRSCs is most suitable for simple configurations (for example, where one network is split into two) or for hosts that rely on another host for gateway support. In the example, HOST11 passes all session requests to HOST21 or HOST22. Those hosts are then responsible for alias name translation, alias address requests, and adjacent SSCP rerouting.

Predefined Cross-Domain Resources with Network Specification

This method of predefining CDRSCs involves including NETWORK definition statements in the CDRSC definition to identify the network in which the resource is located. When a CDRSC definition follows a NETWORK definition statement, VTAM recognizes the specified CDRM as the actual owning SSCP and not as the next SSCP on the session setup path. A session with that CDRM is required only if it is also the next SSCP on the session setup path. The CDRSC definition statement can optionally include the name of the CDRM that owns the logical unit represented by the CDRSC.

In the following example, which is used to define TERM1 in HOST11, TERM1 is defined to be in NET3, and its owning CDRM is HOST31.

```

                VBUILD TYPE=CDRSC,...
                NETWORK NETID=NET3
TERM1          CDRSC CDRM=HOST31,...
```

VTAM recognizes that HOST31 is the actual owning SSCP and not the next SSCP on the session setup path.

Following are some advantages of using this method:

- In a configuration with no name conflicts, this method defines the shortest session setup path. However, unless you are using NQNMODE=NQNAME, a resource, even when defined with a NETID, cannot be predefined if its name is not uniquely known in the host's network. If you are using NQNMODE=NQNAME, a resource, when defined with a NETID, can be predefined even if the name is duplicated in another network.

This method of predefining CDRSCs is designed for those systems that want to quickly locate a resource and use its real name. Therefore, the name must be uniquely known within the network in which it is defined. If the name duplicates a name in an interconnected network, it must be using its network-qualified name.

- Predefining CDRSCs in this way allows you to define very specific adjacent SSCP tables. Because the NETID and, optionally, the destination SSCP are known, you can define adjacent SSCP tables for a specific NETID or NETID and SSCP name. This can eliminate VTAM's need for a default SSCP list, which might not provide the shortest session setup path.
- It adds security by restricting only authorized SSCPs to access resources in this host, assuming no dynamic CDRSCs are permitted.
- You can use NQNMODE=NQNAME to enable CDRSCs in different networks to have the same name.

Following are some disadvantages:

- If you specify the VFYOWNER=YES operand on the CDRSC definition statement, changing CDRMs in a backup situation can be complex. That is, if you have coded the CDRM operand on the CDRSC definition statement, VTAM assumes that it is the owning SSCP name. The operator must issue a MODIFY CDRM operator command in each network that contains a CDRSC definition, with the previous CDRM listed as the owner. Then (as in Figure 113 on page 460), if HOST31 fails and TERM1 is acquired by HOST32 but the CDRM is not updated in HOST11, the session setup attempt fails because HOST31 is still defined as the destination logical unit's owning CDRM. By specifying VFYOWNER=NO on the CDRSC definition statements along the route, VTAM does not verify if that CDRM is the owner but assumes that the session request contains the backup owner and thus routes the session request to that CDRM. Then, only the operators at the session origin need to modify the CDRSC owner to the backup CDRM owner.
- If you are not using NQNMODE=NQNAME, this method does not work for interconnected networks with duplicate names. The method requires that all the connected networks have unique resource names. In many cases, renaming and predefining CDRSCs using NETID or using network-qualified names is preferable to using the alias name translation facility.

This method is recommended for networks that contain unique names and require fast session setup.

If the CDRSC moves randomly among networks, specifying VFYOWNER=NO allows session setup to succeed, but added security is lost in that the intermediate VTAMs on the session setup path do not verify the owning host.

Dynamic Cross-Domain Resources

Instead of predefining CDRSCs, you can let VTAM dynamically allocate and define some or all of them.

Dynamic CDRSC definition is used when VTAM receives a request to establish a session from or for an undefined cross-domain resource or for directory search requests. VTAM handles the request by dynamically creating a temporary CDRSC definition for the cross-domain resource. Without dynamic definition of CDRSCs, the number of resources requiring definition can be extremely large. If you provide networking services within an enterprise or for external users, this method is recommended.

For example, in Figure 113 on page 460, if you choose not to predefine a CDRSC for TERM1 and the CDRM definition for HOST11 and HOST31 is set up to allow dynamic CDRSC definition, the following occurs:

1. A CDRSC for TERM1 is dynamically allocated in HOST11.
 2. Unless you are using network-qualified names, if duplicate resource names exist, VTAM must select an alias name to refer to the session partner located in the other network. For example, if TERM1 exists in both NET1 and NET3, VTAM chooses an alias name whereby NET1 refers to the TERM1 located in NET3. VTAM uses the NetView alias name translation facility or the VTAM session management exit routine to identify the network in which the session partner resides. Then, VTAM resolves the alias name to the name used in the session partner's network. Optionally, the NetView alias name translation facility can be used to obtain the name of the SSCP that owns the session partner.
- If you are using network-qualified names, the NETID uniquely identifies the resources, and VTAM does not have to select an alias name for the session partner located in the other network.
3. VTAM uses an adjacent SSCP table to locate the next SSCP to which the session-setup request should be sent to reach the destination network.

When the session-initiation request leaves the first gateway (that is, the gateway NCP connecting NET1 and NET2), VTAM sends out a search to find the real name and NETID to replace the alias name for the destination logical unit if the alias name has not yet been translated into a real name and NETID. For information on how this affects routing, see "Name Assumption" on page 460.

If no translation takes place in the configuration in Figure 113 on page 460, the session initiation request means that the destination network is assumed to be NET3 when the request is sent to either HOST31 or HOST32.

Following are some advantages of using this method:

- The number of CDRSC definition statements required in each host is greatly reduced. VTAM storage requirements are also reduced because only needed definitions exist.

- After the real name and network are known, the lack of predefined CDRSCs does not affect the length of the session setup path because the dynamic CDRSCs are retained for a user specified time interval (CDRSCTI start option) after the last session terminates.

Following are the disadvantages of using this method:

- The main disadvantage of eliminating predefined CDRSCs is that the length of the session setup path for unqualified session requests is increased before VTAM identifies the destination logical unit's real name and network. In this case, a CDRSC is dynamically allocated using the alias name. VTAM must assume that the name supplied to it in a session request is an alias name. After the real name is determined (by using the alias name translation facility or the alias function of the session management exit routine or from the session initiation response), the acquired information (the real name and network) is merged into the existing dynamically allocated CDRSC. Unless an alias name translation facility that supplies the name of the SSCP owning the session partner is used, each adjacent SSCP leading to the destination network is tried until the session is established or it is determined that none of the SSCPs owns that requested session partner. This process is repeated by each gateway-capable SSCP that receives the session initiation request.
- If an inactive LU is found which represents the destination LU, VTAM can allocate a dynamic CDRSC to enable the session initiation request to be sent to other VTAMs. The decision to allocate the dynamic CDRSC depends on several factors. If the session initiation request specified that the destination LU's owning SSCP was this host, the request fails. Also, if the LU is not eligible to be made into a shadow resource, then the session is rejected.
- An adjacent SSCP table is required if you want to use dynamic CDRSC definition for destination LUs. You can predefine this table or specify the DYNASSCP start option to have VTAM create a dynamic table.

Using this method of dynamically allocating CDRSCs is most beneficial for intermediate and destination networks along the session setup path; the destination logical unit's real name and network identifier are known by the time the session request reaches these hosts, so replacing CDRSCs is not required. Without dynamic definition of CDRSCs, the number of resources requiring definition can be extremely large. If you provide networking services within an enterprise or for external users, this method is recommended.

If the dynamic CDRSC definition is authorized, VTAM:

- Builds a CDRSC minor node definition to represent the undefined LU
- Records the owning CDRM name (when determined)
- Initializes LU information from the session-initiation request or response.

Dynamically defined CDRSC minor nodes are collected in a CDRSC major node named ISTCDRDY. ISTCDRDY is activated automatically during VTAM initialization, and deactivated automatically during VTAM termination. ISTCDRDY can be deactivated and activated by an operator command; this causes all currently active dynamic CDRSCs to be deleted. In addition, all sessions involving dynamically created CDRSCs (including CP-CP sessions with this host, if the partner CP was dynamically defined) are terminated.

While allowing dynamic CDRSC definition for session partners in another domain, you can still restrict access to application programs using the following:

- Session management exit routine (see *VTAM Customization*)
- Session authorization exit routine (see *VTAM Customization*)
- Application program logon exit routine (see *VTAM Programming*).

Dynamically defined CDRSCs are deactivated and deleted by VTAM on a periodic basis if they are not in use. That is, if a CDRSC has no active sessions and has had none for a defined interval of time, the definition is discarded. This interval is set by the CDRSCTI start option of VTAM.

A dynamically defined CDRSC that becomes a shadow resource has all of its sessions moved to the corresponding LU definition. Because of this, it is considered to have no active sessions as soon as it becomes a shadow resource and is then deleted.

If dynamically defined CDRSCs are used, initial session initiation time can be slightly longer unless network-qualified names are used in session initiation requests.

Defining Adjacent SSCPs

This section discusses adjacent SSCP tables that are used for cross-network session requests. For general information about adjacent SSCP tables, see “Adjacent SSCPs” on page 398.

You can use adjacent SSCP tables in a multiple-network environment to:

- Reach destination SSCPs in other networks.
- Determine the next SSCP in the session-initiation path to reach a destination network for an LU-LU session.

Types of Adjacent SSCP Tables

There are three types of SSCP tables that are used for cross-network session requests:

Network-specific list

The network-specific list has a NETWORK definition statement coded in the adjacent SSCP table, and the NETID operand specifies the NETID of the destination network.

Network-specific and CDRM-specific list

The network-specific and CDRM-specific list specifies both a destination network and a destination CDRM.

Default list

The default SSCP list is used whenever one of the following conditions is true:

- The destination network is not known.
- The destination network is known, but there is not a network-specific table associated with the network.
- The destination network and owning SSCP name are known, but the specific table cannot be located.

Adjacent SSCP's

When a CDRSC's network is known, but its owner is not, VTAM uses the network-specific adjacent SSCP table for that network. If none is defined, the default is used.

The default SSCP list is part of the adjacent SSCP table. You can include network-specific SSCP lists for each network and a default list for use when the real network is not known.

Deciding Whether to Code Adjacent SSCP Tables

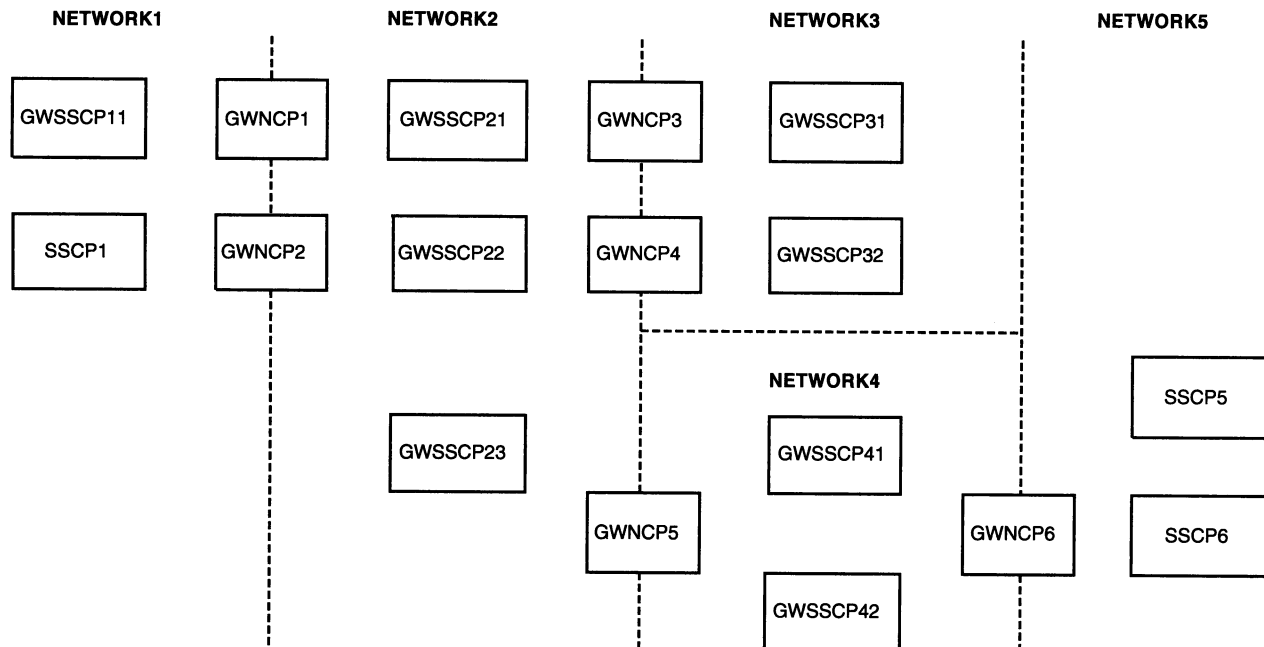
A host does not need an adjacent SSCP list for a network if the following are true:

- The host SSCP is directly in session with the destination SSCP (that is, there are no intermediate SSCPs on the session path), and the host SSCP knows the destination SSCP name.
- Predefined CDRSCs without NETWORK, but with CDRM, are specified.
- An alias name translation program or the alias function of the session management exit routine is used to find the destination SSCP name.

Note: You might need to define adjacent SSCP tables containing the type 2.1 network ID for session requests destined for a non-native network logical unit.

Sample of Adjacent SSCP Tables for a Multiple-Network Environment

The sample adjacent SSCP table definitions included here are written for the configuration in Figure 115.



Note: The connections between hosts and NCPs have been omitted for clarity. You can assume that each NCP connects to each host in the networks it joins.

Figure 115. Multiple-Network Configuration

The adjacent SSCP table in the following example applies to GWSSCP21 in NETWORK2 and contains default SSCP lists for all the networks shown in Figure 115.

TABLE2 VBUILD TYPE=ADJSSCP

```

*****
*  DEFAULT SSCP LIST: THIS LIST IS USED WHEN THE CDRM                *
*  IS UNKNOWN AND NETWORK IS UNKNOWN                                *
*****
      NETWORK                                UNNAMED NETWORK DEFINITION
GWSSCP11 ADJCDRM                            WITH DEFAULT ADJACENT SSCP
GWSSCP31 ADJCDRM                            LIST FOR ALL NETWORKS
GWSSCP32 ADJCDRM
GWSSCP41 ADJCDRM
GWSSCP42 ADJCDRM
*
*****
*  DEFAULT SSCP LIST FOR NETWORK1: USED WHEN THE CDRM IS UNKNOWN    *
*  AND NETWORK IS KNOWN TO BE NETWORK1 *
*****
      NETWORK NETID=NETWORK1  DESTINATION NETWORK
GWSSCP11 ADJCDRM              ADJACENT SSCP FOR NETWORK1
*
*****
*  DEFAULT SSCP LIST FOR NETWORK2: USED WHEN THE CDRM IS UNKNOWN    *
*  AND NETWORK IS KNOWN TO BE NETWORK2 *
*****
      NETWORK NETID=NETWORK2  DESTINATION NETWORK
GWSSCP22 ADJCDRM              DEFAULT ADJACENT SSCP LIST
GWSSCP23 ADJCDRM              FOR NETWORK2
*
*****
*  DEFAULT SSCP LIST FOR NETWORK3: USED WHEN THE CDRM IS UNKNOWN    *
*  AND NETWORK IS KNOWN TO BE NETWORK3 *
*****
      NETWORK NETID=NETWORK3  DESTINATION NETWORK
GWSSCP32 ADJCDRM              DEFAULT ADJACENT SSCP LIST
GWSSCP31 ADJCDRM              FOR NETWORK3
*
*****
*  DEFAULT SSCP LIST FOR NETWORK4 OR NETWORK5: CDRM IS UNKNOWN AND  *
*  NETWORK IS KNOWN TO BE NETWORK4 OR 5*
*****
      NETWORK NETID=NETWORK4  DESTINATION NETWORK
      NETWORK NETID=NETWORK5  DESTINATION NETWORK
GWSSCP41 ADJCDRM              DEFAULT ADJACENT SSCP LIST
GWSSCP42 ADJCDRM              FOR NETWORK4 AND NETWORK5
*
*****
*  CDRM-SPECIFIC LIST FOR SSCP5: CDRM IS KNOWN                      *
*  TO BE SSCP5 IN NETWORK5                                          *
*****
SSCP5    CDRM                                DESTINATION SSCP
GWSSCP42 ADJCDRM                            ADJACENT SSCP LIST
GWSSCP41 ADJCDRM                            FOR SSCP5 IN NETWORK5

```

Example Default Lists

In the above adjacent SSCP table, the default SSCP list for all networks is:

```
GWSSCP11
GWSSCP31
GWSSCP32
GWSSCP41
GWSSCP42.
```

This list is used whenever the following are true:

- The destination logical unit's network is not known.
- The destination network ID is known, but no entries correspond to the destination network.

The default SSCP list for NETWORK4 and NETWORK5 is as follows:

```
GWSSCP41
GWSSCP42
```

This list is used whenever VTAM knows the destination network to be either NETWORK4 or NETWORK5, but does not know the destination SSCP name.

If the destination SSCP is known to be SSCP5 in NETWORK5, the following adjacent SSCP list is used instead:

```
GWSSCP42
GWSSCP41
```

Example Network-Specific Lists

The rest of the adjacent SSCP table is used when VTAM sets up sessions with CDRSCs whose owning CDRM and network are known. For example, if GWSSCP21 has the following definition statement coded in its CDRSC major node:

```
      NETWORK  NETID=NETWORK3
TSO02  CDRSC   CDRM=GWSSCP32
```

Because the CDRM is coded on the CDRSC definition statement, VTAM assumes that this CDRM is the true owner of the resource. Because the NETWORK definition statement is coded, VTAM uses the coded CDRM and the network-specific SSCP table to try to locate the CDRSC.

VTAM first builds an adjacent SSCP table for the CDRSC. The CDRM is defined as SSCP1, so GWSSCP32 is the first entry in the list. The rest of the list consists of the SSCP's that are adjacent to GWSSCP32, as defined in the default portion of the sample table:

```
GWSSCP32 CDRM
GWSSCP31 ADJCDRM
```

If the session setup fails through GWSSCP32, VTAM routes the request through GWSSCP31.

If no network had been coded for TSO02, VTAM would have used the default SSCP list (GWSSCP11, GWSSCP31, GWSSCP32, GWSSCP41, GWSSCP42). This could result in some session setup requests being routed outside the network through the gateway SSCP's. The adjacent SSCP table can thus prevent time and

resources from being used in a search that might be unnecessary. The session management SSCP selection function can also be used to alter the SSCP list.

For VTAM V3R2 and later releases, an ADJSSCP table defined with a NETWORK definition statement for the host network is used only when the destination resource is known to reside in the host network. If a destination resource is known to reside in NETWORK2, the list of ADJSSCPs used for a session request is:

GWSSCP22
GWSSCP23.

Prior to VTAM V3R2, an ADJSSCP table defined with a NETWORK definition statement for the host network is treated as a default table.

Overriding the SSCP Lists

If a list of adjacent SSCP's exists for a destination network, an SSCP, or both and if VTAM already has an SSCP-SSCP session with the destination SSCP, session setup processing can override the list; that is, the destination SSCP becomes the first choice for the specific session request. This override is temporary and does not change the actual adjacent SSCP table that you define.

Thus, the default SSCP list (TABLE2) includes the following definition statements:

NETWORK	NETID=NETWORK3	DESTINATION NETWORK
GWSSCP32	ADJCDRM	DEFAULT ADJACENT SSCP LIST
GWSSCP31	ADJCDRM	FOR NETWORK3

Assume that a cross-network SSCP-SSCP session exists between GWSSCP21 and GWSSCP31. If GWSSCP21 receives a cross-network Initiate request for a session with a logical unit owned by GWSSCP31 in NETWORK3, the list of adjacent SSCP's in the table (GWSSCP32 and GWSSCP31) is reordered for this session setup as follows:

GWSSCP31
GWSSCP32.

The actual table you define is unaltered.

The destination SSCP name, GWSSCP31, is available in the Initiate request if the alias name translation facility or a predefined CDRSC with CDRM name is used.

Request Routing

Session initiation and INQUIRE APPSTAT requests are routed from the originating LU's SSCP to the destination LU's SSCP using trial-and-error routing. There are several different means of determining the list of adjacent SSCP's to try when performing trial-and-error routing. However, after the list is determined, the processing is very similar for each of these RUs.

Rerouting of Requests

If the adjacent SSCP to which a cross-network session setup request is routed does not own the destination logical unit, it reroutes the request, provided that the following are true:

- An appropriate adjacent SSCP is available.
- The maximum SSCP rerouting count has not been reached.
- This SSCP is gateway capable.

Otherwise, the request is rejected.

If the SSCP that sent the session setup request is in the list, the entry is not used to reroute the request. This prevents a request from looping. The actual table you define is unaltered.

Rules for Cross-Network Session Request Routing

The rules for routing cross-network session requests are as follows:

- The request is not routed if its visit count (SSCP rerouting count) has reached 0.
- The request is never routed back to the SSCP from which it was received.
- The request is never routed back to its originating SSCP.
- The request is never routed in this network if it was received from an SSCP in this network.
- An SSCP receiving a session request from another network can route the request to a different SSCP in that network through a different gateway NCP. It cannot route the request back using the same gateway NCP.
- The request received by a non-gateway SSCP cannot be rerouted by that SSCP.
- The cross-network request received by a gateway-capable SSCP can be rerouted by that SSCP.
- A non-gateway SSCP can perform trial-and-error routing for a session that it initiated.
- If the real NETID of the DLU is known, routing is not attempted to a non-gateway SSCP with a different NETID, unless the adjacent SSCP supports the non-native network connection function.
- A request is not routed to an SSCP that does not support the required function. (For example, automatic logon sessions are not routed to SSCPs that do not support automatic logon sessions).

Dynamically Defined CDRSCs and Adjacent SSCP Tables

If dynamic definition of resources is allowed and if no predefined CDRSC is located, VTAM automatically creates a CDRSC for the logical unit as part of routing. After the owning SSCP is found by trial-and-error routing of the session initiation request, the dynamically defined CDRSC is updated with the real network ID and owning SSCP name.

For releases of VTAM prior to Version 4, if two hosts attempt to establish a cross-network session with a logical unit that is not predefined in either of the initiating hosts, a conflict can arise in any intermediate host that receives these requests. The receiving host will then issue a sense code indicating duplicate DLU names. The sessions can be successfully established if the requests can be rerouted. This situation can be avoided, however, by predefining the DLU with NETID or by using the NetView alias name translation facility or the alias function of the session management exit routine to define the name of the resource instead of allowing it to be dynamically defined.

Alias Name Translation and Adjacent SSCP Tables

The NetView alias name translation facility can be used to determine the owning SSCP for a requested resource in the same network. To use this facility for an LU, define the LU and its owning SSCP to the alias name translation facility, because the alias name translation facility cannot translate a name as known in a network to a different name in the same network.

Define each alias name to be the same as the real name. During a session setup request, if VTAM does not have a definition of a destination logical unit or does not know the owning SSCP for the resource, VTAM can invoke the alias name translation facility with the name of the logical unit. The alias name translation facility returns the name of the SSCP defined as the owner of the logical unit, as well as the destination network identifier and the logical unit's alias name in that network. Because this system has only one network, the network identifier is VTAM's network identifier, and the alias name is the same as the real name. Thus, the alias name translation facility is used as a directory to locate the owning SSCP and can direct the session setup request to the proper VTAM.

Cross-Network Routing

In a multiple-network environment, application programs and terminals managed in one network can access SNA resources controlled and managed in another network, but each network can preserve its autonomy and maintain unique network characteristics, such as the network address structure and network naming conventions.

Network Address Structures

When SNA networks are interconnected, each can maintain its current network addressing scheme. However, there are some addressing considerations.

Maintaining Address Structures

The network addresses assigned to nodes (host processors or communication controllers) in one network can be duplicated in other connected networks, as can the following subarea addressing capabilities:

- Extended subarea addressing
- Extended network addressing
- Non-extended network addressing (MAXSUBA).

The maximum number of addressable network nodes or subareas do not have to be consistent or compatible across the interconnected networks.

SNA network interconnection (SNI) eliminates the architectural requirement that interconnected networks must use a common network address structure with all unique network addresses. Instead, it permits each network to establish its own network addressing capabilities.

Addressing Considerations for Interconnected Networks

Any network that has greater addressing capability than its adjacent networks must have extended subarea addressing-capable gateway NCPs and gateway SSCPs that are involved in setting up cross-network sessions for high subareas. Specifically, consider the following:

- The gateway NCP's addressing capability in each network must be at least equal to the largest subarea address in the network, and its subarea address in the network must be within the range supported by the subarea with the lowest addressing capabilities. This restriction applies only for subareas that are using the gateway NCP for cross-network sessions.
- The gateway VTAM's addressing capability must be greater than or equal to the largest subarea address in the network, and its subarea address must be within the range supported by the SSCP with the lowest addressing capabilities. That is, the gateway SSCP must be capable of supporting a CDRM session with any SSCP in the network, and it must be capable of owning the gateway NCPs it is to control.
- To properly support communication between networks where either network contains subareas that support cross-network sessions and that have subarea addresses above the capabilities of any of the other subareas involved in supporting or setting up the cross-network session, the gateway SSCPs in each network must be capable of supporting extended subarea addressing.
- For an extended subarea addressing gateway NCP to successfully send a CONTACT request unit to an adjacent cross-network gateway NCP with a subarea address greater than 255, the link station must be owned by an extended subarea addressing-capable VTAM.

Network Naming Conventions

SNA requires each terminal and application program (logical unit) to be identified by a unique, network-qualified name. Network-qualified names are used by specifying the NQNM=QQNAME start option or defining cross-network resources with the NQNM=QQNAME operand on the CDRSC definition statement.

If you choose not to use network-qualified names, you can use the optional facility SNI provides for translating terminal and application program names to cross-network (alias) names to avoid name duplication conflicts. By using this facility, the name structure used in each network can be maintained. A common network naming convention among the interconnected networks does not have to be established. If multiple SNA resources are known by the same name in different networks, the name conflict can be resolved by using one or a combination of the following:

- The NetView alias name translation facility
- The VTAM session management exit routine
- Network-qualified names.

If you choose to fully implement network-qualified names, you do not need to use either the NetView name translation facility or the VTAM session management exit routine to resolve duplicate names.

Controlling Paths for Interconnected Networks

For SNA network interconnection (SNI), paths are defined the same as for single- and multiple-domain environments within each network (see “Network Routing for Subarea Nodes” on page 151 and “How to Plan Routes in Your Network” on page 418). Rules for explicit and virtual routes do not cross network boundaries. For example, in Figure 116 assume that you are setting up a session between LU1 and LU2 and that they are in the same network. The same explicit route that is used from HOST1 to GWNCBPB (ER0) would have to be used from GWNCBPB to HOST2. However, because HOST1 and HOST2 are in different networks, the two explicit route definitions are independent of each other.

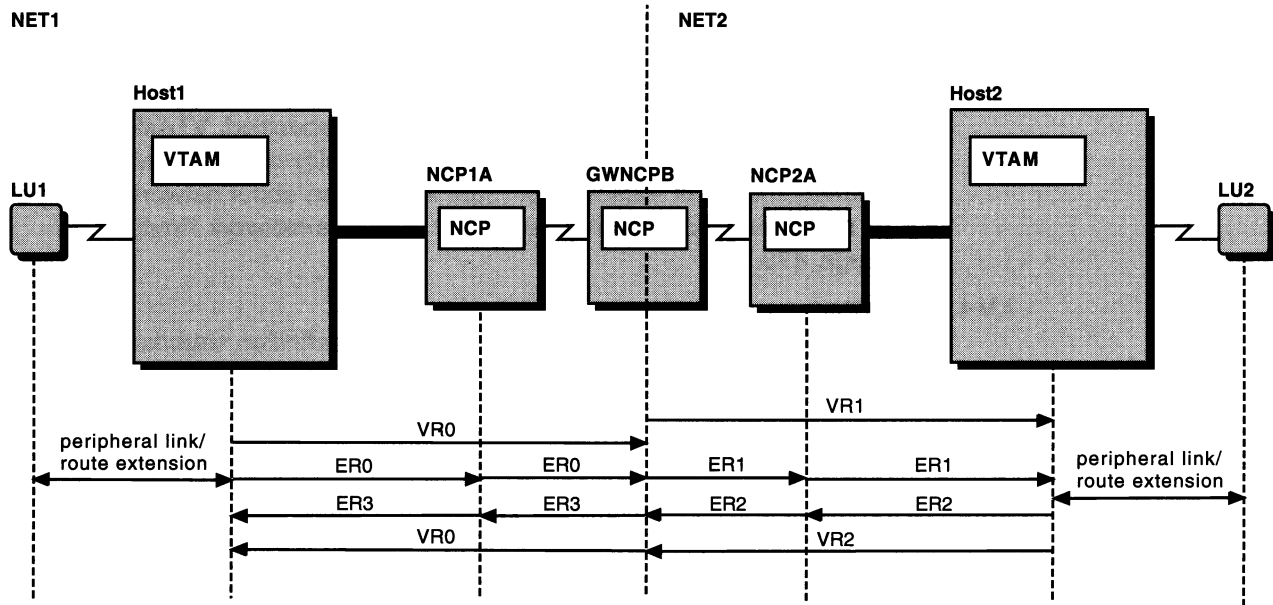


Figure 116. Multiple-Network Paths

Paths do not cross network boundaries. Cross-network routing requires separate paths in each network, each having the gateway NCP as the endpoint. If you are combining networks, you might have to add paths in non-gateway NCPs that go to the gateway NCP.

If you are splitting an existing network into two networks, redefine the paths so that each path passing through a gateway NCP is defined separately in each of the networks.

Gateway Path Selection

For cross-network sessions, the gateway path selection function of the session management exit routine can be used to modify a list of alternate gateway paths to distribute LU-LU sessions among two or more gateway NCPs. You can shorten the list by deleting entries or reorder it by putting the preferred paths at the top of the list.

If you do not provide this function in your session management exit routine, VTAM uses the list of gateway paths determined by the GWPATH definition statements. However, the gateway NCP used for the SSCP-SSCP session is always the first one used for LU-LU sessions. For information about using the session management exit routine, see “Session Management Exit Routine” in *VTAM Customization*.

A gateway path is selected by one of the following methods:

SSCP-SSCP sessions

Each successive gateway path between two VTAMs is tried until either the SSCP-SSCP session is established or all of the possible session paths have been tried unsuccessfully.

The class-of-service name that is normally used is the COS name for SSCP sessions, ISTVTCOS. This is the default name that is used for establishing a cross-network SSCP-SSCP session and, if used, is the COS name that is passed to the NetView alias name translation facility or the VTAM session management exit routine. You can specify a different class-of-service name by coding the value on the ADJNETCS operand on the GWPATH definition statement. VTAM uses that COS name to locate the virtual route list in the proper COS table as specified on the COSTAB operand of the BUILD or NETWORK definition statements in NCP. If the ADJNETCS operand is specified, VTAM does not invoke the NetView alias name translation facility or the VTAM session management exit routine. For more information about network routing in a multiple-network environment, refer to "Cross-Network Routing" on page 473.

LU-LU sessions

Each gateway NCP (defined by a GWPATH definition statement) between the two VTAMs is tried until one of the following occurs:

- The requesting logical unit receives a positive response to the Initiate request for the session, meaning that the requested session partner has been located.
- All the possible gateway NCPs (defined by the GWPATH definition statements for the CDRM in another network) have been tried unsuccessfully.
- A failure not related to the GWNCP is detected.

If multiple gateway paths are used, VTAM selects a particular gateway NCP for each LU-LU session. The gateway NCP used for the cross-network SSCP-SSCP session is always the one VTAM attempts first for LU-LU session setup. If this is unsuccessful, the gateway VTAM selects another gateway NCP for the session. If you coded a session management exit routine to shorten or reorder the list of gateway NCPs, the gateway VTAM uses that altered list to establish the LU-LU session path.

During session establishment of a cross-network LU-LU session, the gateway VTAM that is controlling gateway path selection gets the list of alternate gateway NCPs defined by the GWPATH definition statements. If the BIND RU that flows between the requesting logical unit and the session partner fails because a virtual route required by the session cannot be activated, no further gateway path selection is attempted. VTAM messages are issued to inform the operator that session establishment failed, and the requesting logical unit is informed.

Coding the Gateway Path

The GWPATH definition statement is required for all gateway VTAMs that initiate cross-network SSCP-SSCP sessions. If you do not code GWPATH definition statements following a CDRM definition statement for a VTAM in another network, code the SUBAREA operand on that CDRM definition statement so that VTAM can determine the gateway NCP to use.

Dynamic cross-domain resource definition must be allowed for the host CDRM if the NetView alias name translation facility or the alias function in the session management exit is used to translate logical unit names.

Handling Class-of-Service Tables

Following are several important topics to consider when using COS tables for interconnected networks:

- Using COS tables for gateway VTAMs and gateway NCPs
- Using multiple identical COS tables (that is, using one COS table for more than one network or for more than one subarea node within a network)
- Using different COS tables for interconnected networks
- Using conflicting COS table names.

You can define different COS tables for VTAM to use within its own network or within gateway NCPs that it owns. When the PLU is in VTAM's subarea, VTAM uses ISTSDCOS to resolve the class-of-service name to a virtual route list. The virtual routes originating in the host subarea extend to the SLU's subarea node or to a gateway NCP providing a path to the SLU.

When VTAM is acting as a gateway VTAM that has been designated to resolve COS names for a gateway NCP, VTAM uses the COS table named on the NETWORK or BUILD definition statement in the gateway NCP's generation definition (whichever applies to the adjacent network to which you are routing). The gateway VTAM passes the virtual route list to the gateway NCP, and the gateway NCP performs the route activation for one of the routes in the resulting virtual route list within that network.

To establish a cross-network SSCP-SSCP session, VTAM uses the ISTVTCOS class-of-service name to establish a route to the adjacent network SSCP. On the GWPATH definition statement that identifies the gateway NCP to be used for the session, you can specify the ADJNETCS operand to identify a COS name other than ISTVTCOS. VTAM uses this COS name to determine the specific route in the adjacent network for the session.

All of the COS tables to be used for gateway NCPs controlled by a gateway VTAM must be available to that SSCP. If VTAM attempts to load a COS table that it cannot find, session requests associated with the table fail.

If more than one gateway VTAM shares control of a gateway NCP, those gateway VTAMs can be defined in such a way that some of them allow the others to resolve the COS name. This is specified by the GWCTL operand on the PCCU definition statement. In this case, only those gateway VTAMs responsible for resolving COS names for that gateway NCP must have the COS tables stored in their hosts.

For example, in Figure 117 on page 478, if Host1 has GWCTL=ONLY coded on the PCCU definition statement for GWNCPB, then Host1's COS table is used for an SSCP-SSCP session between Host1 and Host2. If Host1 and Host2 have GWCTL=SHR coded on the PCCU definition statements for GWNCPB, the originating VTAM's COS table is used.

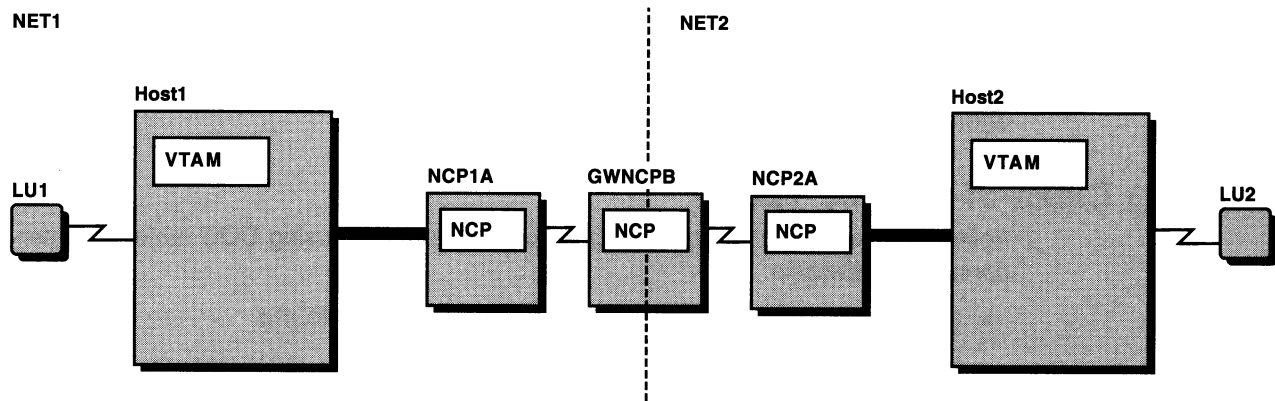


Figure 117. COS Resolution in a Multiple-Network Environment

The same COS table can be used for more than one network. You can also create a universal COS table to be used for all networks. This is possible if the COS names are the same and if virtual routes originating in different subarea nodes have identical VR numbers for routes providing the same levels of service.

If the same COS table can be used in more than one network connected to a gateway NCP, code the same table name on the COSTAB operands for the BUILD or NETWORK definition statements representing those networks in the NCP major node. VTAM loads only a single copy of the table into VTAM's storage.

If your interconnected networks require different COS tables, you can associate these table names with their proper networks by coding the table names on the COSTAB operands of the appropriate BUILD or NETWORK definition statements in the NCP generation definition.

The COS table for routes originating in a VTAM host must be named ISTSDCOS. If the same COS table can be used for a gateway NCP that is controlled by a different VTAM host, the COS table must be stored in the other VTAM host under a name other than ISTSDCOS (unless that VTAM can also share the same COS table for routes originating in that host).

In Figure 118 on page 479, networks NETA and NETB are connected in a back-to-back configuration.

The session between the PLU in NETA and the SLU in NETB uses the following three routes:

- A route within NETA from the PLU's subarea to the gateway NCP's (GWN1's) subarea in NETA
- A route from the NETX subarea within GWN1 to the NETX subarea within GWN2
- A route from the NETB subarea within GWN2 to the SLU's subarea.

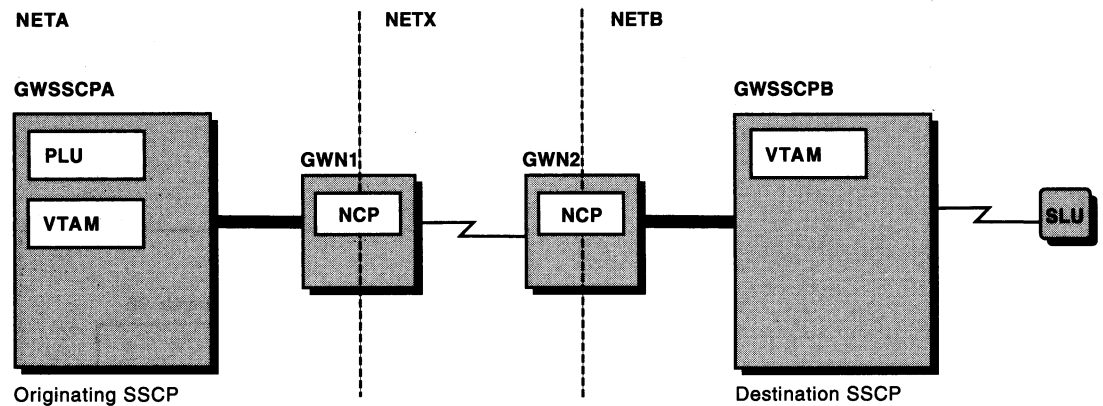


Figure 118. COS Tables and Routing in an SNI Back-to-Back Configuration

Assuming that GWN1's resources reside in NETA (that is, NETA is GWN1's native network), GWN1's generation definition contains the following:

```
BUILD    NETID=NETA,...,COSTAB=COSNETA1
:
NETWORK NETID=NETX,...,COSTAB=COSNETX1
```

Also, assuming that GWN2's native network is NETB, GWN2's generation definition contains the following:

```
BUILD    NETID=NETB,...,COSTAB=COSNETB2
:
NETWORK NETID=NETX,...,COSTAB=COSNETX2
```

The following COS tables are used:

- In GWSSCPA, the SSCP of the PLU uses ISTSDCOS to select the list of virtual routes in NETA from which a route is selected for the session.
- GWSSCPA, as the gateway VTAM controlling GWN1, resolves the COS name for NETX to a list of virtual routes in NETX. It uses the table COSNETX1.
- GWSSCPB, as the gateway VTAM controlling GWN2, resolves the COS name for NETB to a list of virtual routes in NETB. It uses the table COSNETB2.

VTAM uses its own default virtual route list when a cross-network SSCP-SSCP session activation is attempted with a COS name other than ISTVTCOS or the unnamed class of service. Session activation proceeds, but a message is sent to warn the operator that the COS table should be corrected. Whenever convenient, you should correct the problem by restarting VTAM.

Note: For user session requests, you can specify a substitute set of COS parameters that are used if the COS name is unknown to this VTAM. See "How Session Traffic Is Assigned to a Specific Route" on page 414.

Address Translation

A gateway NCP uses one subarea for each network that is to be connected. Thus, a gateway NCP appears to reside in more than one subarea. Associated with that subarea is a pool of element addresses that can be used for cross-network sessions. When a cross-network session is requested, the gateway NCP assigns a pair of element addresses, one in each network on either side of the gateway NCP,

to the session partners. This enables the gateway NCP to create within each network a representation of the session partner in the other network.

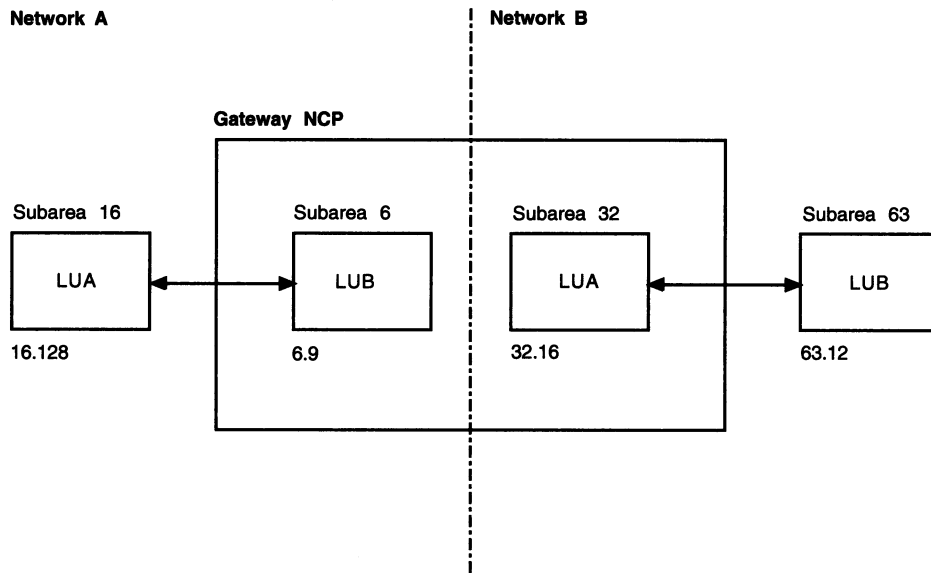


Figure 119. Address Translation

For example, Figure 119 shows two networks (networks A and B) that are connected through one gateway NCP. An LU in Network A (LUA) is in session with another LU in Network B (LUB). LUB has a subarea address of 63 and an element address of 12 in its own network. Because these addresses might not be valid in Network A, the gateway NCP has given LUB a subarea number of 6 (the subarea of the gateway NCP in Network A) and an element address of 9. As a result, LUB now has a valid address in Network A that can be used for routing session data. When an LU in Network A sends data to LUB, it is routed to subarea 6, element 9, in the gateway NCP. The gateway NCP translates this address into the address to be used for LUB in Network B. It then passes the session data over to Network B. The session data is then routed through Network B to LUB in the usual manner.

Similarly, the gateway NCP has given LUA an address in Network B. Any session data destined for LUA from Network B is sent to this address.

Element Addresses

Address translation is also done for cross-network SSCP-SSCP sessions. Each LU and SSCP with which you want to have a cross-network session requires at least one element address. The number of addresses that can be used to represent cross-network resources within a given network depends on the following:

- If extended network addressing is used, 32768 element addresses can be used to assign alias addresses.
- If non-extended network addressing is used, then the element pool is determined by the MAXSUBA specification. For example, if a network has MAXSUBA=63, each subarea can have at most 1024 element addresses.

Resources attached to the gateway NCP require element addresses in the native network. Therefore, while the number of possible cross-network sessions is not equal to the maximum number of elements in each subarea, the two must be considered together.

Parallel Sessions and Element Addressing

Application programs that use non-parallel sessions require only one address to represent the primary LU to all of the secondary LUs. However, application programs that use parallel sessions require a unique address to represent the primary LU to each one of the secondary LUs. If you plan to have cross-network parallel sessions, the network containing the secondary LU must have one element address for each parallel session with the primary LU. In the network of the primary LU, only one element address is required for all sessions with the secondary LU. Therefore, the number of addresses you should reserve in any particular network to represent application programs in other networks depends on whether the application program requires a unique primary LU address for each of its sessions.

For example, if terminals in your network are logged on to an application program such as CICS, you need only one element address in the gateway NCP to represent that CICS to your network. If parallel sessions are established between your CICS system and the CICS in the other network, the PLU must use an address for each session. If your CICS is acting as the SLU for these parallel sessions, an element address is required for each session in the gateway NCP to represent the PLU to your network. If the CICS in the other network is acting as the SLU for these parallel sessions, only one element address is required in the gateway NCP to represent that CICS as the SLU to your network.

The gateway NCP needs an additional address for the other network to represent each of your CICS parallel sessions to the other network. The gateway NCP also needs element addresses to represent your terminals to the network containing the other CICS.

Multiple ACBs and Element Addressing

If terminals in your network are logged on to an application program that opens a separate ACB for each user (such as the NetView or **MVS** TSO programs), the number of element addresses that you need to represent the NetView program in your network is equal to the maximum number of terminals in your network that are concurrently logged on to that application program. In addition, the network containing the application program must have an element address for each of your terminals.

Resource Name Translation

SNA requires that each network resource be identified by a unique name. This requirement also applies to a multiple-network environment, unless you are using network-qualified names. Network-qualified names are implemented by specifying the NQNM= NQNAME start option or defining cross-network resources with the NQNM= NQNAME operand on the CDRSC definition statement. If you do not use network-qualified names, cross-network resources cannot be identified by the same name. For example, a problem occurs when a logical unit in one network requires access to a resource in another network that uses the same name to identify a different logical unit.

The class-of-service (COS) name and the logon mode entry name are also important when establishing sessions. The same COS and logon mode entry names must be used in each of the interconnected networks to establish cross-network sessions.

To meet these naming requirements when you are connecting networks, do one of the following:

- Redefine any duplicate resource names to preserve name uniqueness, and ensure that you use the same COS and logon mode entry names as the attached networks. (If you are using network-qualified names, you do not need to redefine resources to resolve duplicate names.)
- Define any resources with duplicate names so that no more than one of them is specified (either by definition or by start option) with NQNMODE=NAME and the others are specified (either by definition or by start option) with NQNMODE=NQNAME.
- Use the VTAM alias selection function in the session management exit routine to translate any duplicate resource name to unique resource names and map unlike COS and logon mode entry names in one network to the proper equivalents in the other network. For details about the session management exit, see “Session Management Exit Routine” in *VTAM Customization*. (If you are using network-qualified names, you do not need to use the session management exit routine to resolve duplicate names.)
- Use the NetView alias name translation facility to translate any duplicate resource names to unique resource names and map unlike COS and logon mode entry names in one network to the proper equivalents in the other network. (If you are using network-qualified names, you do not need to use the NetView alias name translation facility to resolve duplicate names.)

The resource names that can be translated by the NetView alias name translation facility or in the session management exit routine are:

- Logical unit names
- Logon mode names
- Class-of-service names.

Alias Selection Function of the Session Management Exit Routine

You can use the alias function of the session management exit routine to perform name translation. By using this exit routine, you can provide your own alias and real names for logical units, logon mode names, class-of-service names, and SSCP names, instead of using the NetView alias name translation definitions. You can also indicate whether the NetView alias name translation facility is to be invoked for additional translation.

The session management exit routine can be invoked to translate a real or alias logical unit name associated with one network to a real or alias logical unit name for the destination network. You can map different LOGMODE and COS names in different networks to be functionally equivalent, and can also provide directory services by providing the owning SSCP names for the destination resource.

If you are using the NetView alias name translation facility, you can selectively invoke it to perform additional translations not performed in the exit routine. When the exit is invoked, there is an indicator that informs you if the NetView alias name translation facility is currently active. The exit routine can determine whether the session requests should be rejected, depending on the need for further translation by the NetView alias name translation facility and on whether it is active. It can also accept or reject the use of a substitute COS.

For information on using the alias selection function of the session management exit routine, see “Session Management Exit Routine” in *VTAM Customization*.

NetView Alias Name Translation Facility

The NetView program provides the alias name translation facility that can do the following:

- Translates any LU name duplication conflicts
- Maps logon mode and COS names.

Translating LU Names

No two LUs in an SNA network can have the same name. When you connect independent SNA networks, however, there might be some duplicate names among these networks. To provide the unique LU names required for SNA sessions but still allow each network to have independent resource names, interconnection provides for name translation between networks. Logical unit names are translated from names known in one network to names known in the network of the session partner to avoid duplicate names.

Network-qualified names provide an option to name translation and enable you to have resources by the same name in different networks without requiring you to use name translation.

If you are not using network-qualified names, you can use name translation, which allows each LU to have two names: a real name by which it is known in its own network and an alias name by which it is known in another network. For example, an LU in NETA is known by its real name LUA1A3 in NETA, and an LU in NETB is also known by its real name LUA1A3 in NETB. If NETA is connected to NETB, a name duplication conflict develops. The alias name translation facility enables LUA1A3 in NETA to be identified by an alias name in NETB (for example, BBA1A3). By using this procedure, you can maintain name uniqueness within a network and adhere to SNA naming requirements.

The NetView alias name translation facility uses installation-defined translation tables. The tables are a one-to-one mapping of the alias LU name to the real LU name and their corresponding networks. This technique differs somewhat from that used for maintaining alias addresses across a network boundary. That is, an exact one-to-one mapping is defined by an installation using the translation tables. However, terminal users and network operators do not normally know an LU by its network subarea and element address. Instead, it is known by its unique name in the network. If the alias name (by which an LU is known) is able to change dynamically with each session, network operations might be very difficult, and network security might be affected. The NetView translation tables therefore predefine the relationship between an alias name and a real name.

Note: The alias name translation facility cannot translate a name as known in one network to a different name in the same network. Also, because NetView domain identifiers cannot be translated, they must be unique for all NetView programs that have cross-network sessions. Resource names for LU 6.2 sessions also cannot be translated.

Mapping Logon Mode Names

Before a session is established, both session partners must agree on a common set of session parameters. They do this by specifying a logon mode name that identifies a particular set of session parameters. Within an SNA network, both session partners use the same logon mode name, thus ensuring that the correct session parameters are used. When you connect independent SNA networks, however, a given set of session parameters might be known by different names in different networks. Also, a given name might refer to different session parameters in each network. The alias name translation facility can be used to map a logon mode name in one network to another name in the destination network that provides equivalent session parameters. This translation can occur in any SSCP on the setup path, after the network identifier of the destination network is determined.

As usual, the logon mode name that is used to establish a session must exist in the logon mode table of the secondary logical unit's SSCP.

Mapping Class-of-Service Names

Before a session is established, both session partners must agree on a common class of service that defines a particular ordered list of virtual routes. They do this by specifying a class-of-service name. Within an SNA network, both session partners can use the same class-of-service name, thus ensuring that the correct class of service is used. When you connect independent SNA networks, however, a given class of service might be known by different names in different networks, or the same name might be used for different classes of service. When the alias name translation facility is used, a class-of-service name in one network can be mapped to a different name in each adjacent network along a cross-network session path to provide equivalent service. This translation can occur in any SSCP on the setup path. For SLU initiated sessions, you must also map the COS name for the origin logical unit (OLU) to the destination logical unit (DLU).

The COS name entry used for the session must exist in the COS table of the primary logical unit's SSCP and in the gateway VTAM that performs the SETCV. For information on determining which gateway VTAM performs the SETCV, see "Cross-Network Routing" on page 473.

The NetView alias name translation facility can be used to establish equivalence for the required session parameters. The NetView alias name translation facility maps the logon mode name in one network to the logon mode name used in the other network. The same process is used for the COS name. If the COS name needed to establish a virtual route and transmission priority in one network does not exist in the COS table of the PLU in the other network, the NetView alias name translation facility can be used.

This alias name translation facility maps the COS name in one network to the COS name in the other network that provides a network route with equivalent characteristics (for example, high speed, high availability, security).

A substitute COS entry can also be used. The substitute COS entry, chosen by specifying the SUBSTUT operand on the COS macroinstruction for one COS entry, indicates the COS entry to be used when the COS specified in a session request is unknown. For more information on how the substitute COS works, see "Substituting Class-of-Service Parameters" in "How Session Traffic Is Assigned to a Specific Route" on page 414.

Defining Alias Names

When defining resources in interconnected networks, you must reconcile any name conflicts among the connected networks. Following are the ways you can do this:

- Adopt a naming convention in all networks that ensures that all names are unique.
- Define alias name translation tables.
- Use the alias function in the session management exit routine.
- Implement network-qualified names.

The method that is best for you depends on how many names must be changed and how much control you have over the various networks.

To use an alias name translation facility, define all the logical units in the interconnected networks that are not already defined and their owning SSCPs to the NetView alias name translation facility and assign alias names. Then, if VTAM does not have a complete definition of a destination logical unit for a session setup request, it invokes the NetView alias name translation facility with the name of the logical unit. The NetView alias name translation facility returns the name of the SSCP defined as the owner of the logical unit, as well as the destination network ID and the logical unit's real name in that destination network.

If the following conditions are true, the request for the session is sent to the owning SSCP:

- No predefined CDRSCs are used.
- VTAM has a session with the owning SSCP.
- Dynamic definition of cross-domain resources is allowed.

VTAM automatically creates a CDRSC definition for the logical unit.

If you have not installed a name translation facility, or if the name translation facility does not return the name of the owning SSCP, SSCP selection using the adjacent SSCP table is tried next.

Note: VTAM uses real names and alias names when referencing an LU. An alias name is the name of the LU as known in the network of its session partner. Because one LU can be in session with partners in more than one network, an LU can have multiple alias names. VTAM can obtain real and alias names through an alias translation application program (for example, NetView). If such an application program or the alias function of the session management exit routine is not in use, VTAM uses network qualifiers to define the real and alias names for logical units, in which case, all resource names must be unique. Therefore, no two LUs can have the same name, whether or not they are in the same network. However, if you are using network-qualified names, you can have duplicate resource names in different networks.

In general, when VTAM receives a session setup request and the alias name translation facility is active, VTAM invokes the alias name translation facility to translate LU names, class-of-service names, and the logon mode name. You can use the TRANSLAT start option to limit the scope of the translation functions for which VTAM invokes the NetView program.

The default for the CDRDYN start option is YES, enabling dynamic CDRSC definition. The CDRDYN start option overrides the CDRDYN operand on the host CDRM definition statement. CDRDYN must be YES for the VTAM in which the NetView alias name translation facility or the alias function of the session management exit routine is used.

If an alias name is defined to VTAM with a CDRSC definition (either predefined or dynamically defined), the DISPLAY operator command can be used to display the real name associated with that alias name (if a session exists with the resource and if VTAM has determined the real name).

You should have a convention for creating alias names. This alias naming convention should identify a name as an alias name. An operator often needs to know whether a resource name is an alias name or a real name. For example, the NetView program requires that the operator enter the real name of a resource when a display is requested.

Note: The generic name used by the alias function should not also be used by USERVAR.

As an example of name translation, Figure 120 on page 486 shows a gateway NCP connecting two networks, NETA and NETB. Both networks contain an application program named IMS. A terminal user, TERMU, in NETA wants to log on to the IMS that is in NETB.

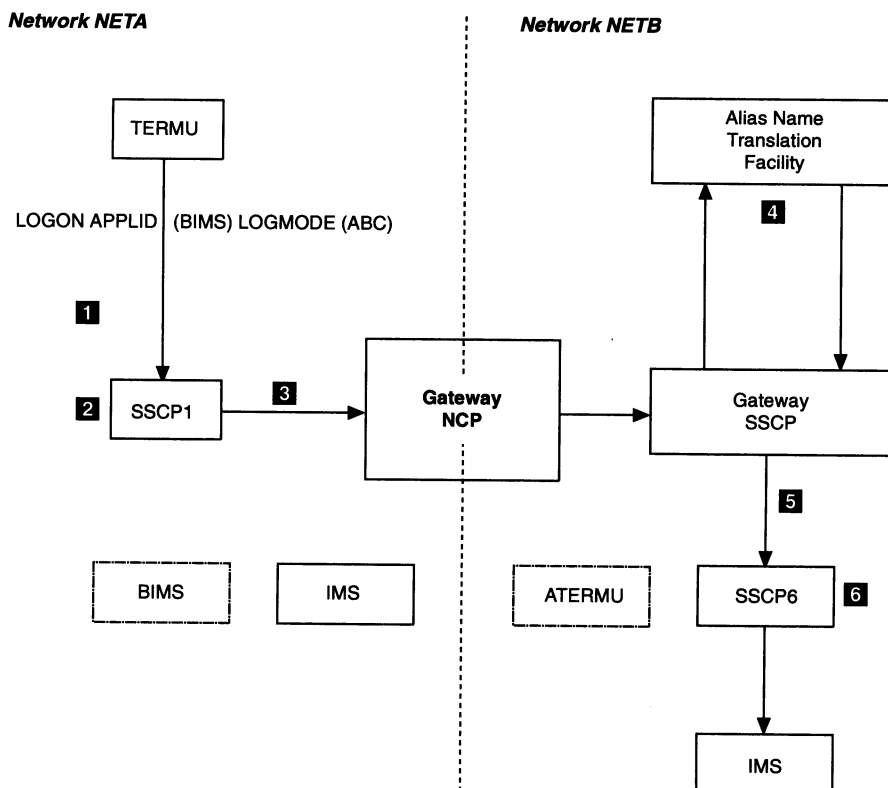


Figure 120. Example of Name Translation

The following steps show how name translation is used in setting up a session between TERMU in NETA and IMS in NETB. The numbers shown in Figure 120 correspond to the following steps.

1. TERMU enters a LOGON command specifying BIMS, which is an alias name for IMS in NETB.
2. The LOGON command is processed by SSCP1, the SSCP that owns TERMU. The COS name FAST is obtained from the logon mode entry named ABC in the logon mode table associated with TERMU.
3. SSCP1 has a dynamic or predefined CDRSC definition without NETID for the resource named BIMS, which indicates that BIMS is owned by the gateway SSCP. SSCP1 therefore passes the request to the gateway VTAM.
4. Because the NetView alias name translation facility is active or because an alias function in the session management exit routine is used, the gateway VTAM supplies the following names for translation:
 - Origin network ID (NETA)
 - Origin LU (TERMU)
 - Alias name of destination LU (BIMS)
 - COS name (FAST)
 - Logon mode name (ABC).

What NetView or the session management alias function returns to the gateway VTAM depends on what user defined name translations are coded in the NetView program or are available from the session management alias function. In this example, the following names are returned:

- Destination network ID (NETB)
 - Real name of destination LU (IMS)
 - Name of owning SSCP for IMS (SSCP6)
 - Alias name for TERMU (ATERMU)
 - Equivalent COS name (QUICK)
 - Equivalent logon mode name (XYZ).
5. A dynamic CDRSC definition is created in SSCP6 for TERMU using the alias name and is associated with the real name

After the session is set up, it appears to SSCP1 that TERMU is in session with a CDRSC in NETA named BIMS. To SSCP6, it appears that IMS is in session with a CDRSC in NETB called ATERMU. However, if SSCP1 or SSCP6 were a VTAM V2R2 or VTAM V3 with the NETID start option specified, SSCP1 or SSCP6 would also be aware that this is a cross-network session.

The rules for name translation are as follows:

- Logical unit names and logon mode names are translated only once. If class of service is provided on the session-setup request, it is translated once from the name known in the origin network to the name known in the destination network. As the response to the session setup request flows back to the origin host, it is translated in each network to determine the COS name as it is known in the adjacent network.
- If the destination LU name must be translated, at least one of the gateway VTAMs that control the first gateway NCP in the session setup path must have an alias name translation facility.

Translation of the destination LU name is attempted in every gateway or gateway-capable SSCP that has an active name translation facility until the names are translated or until a gateway VTAM requests alias addresses from a gateway NCP. A gateway VTAM requests alias addresses if it has designated

control of the gateway NCP or if it is the last gateway VTAM on the first gateway NCP. At that point the destination LU name is assumed to be real if it has not been translated.

- After the destination logical unit real name and network identifier are determined, VTAM still tries to translate the origin LU name at every gateway or gateway-capable SSCP along the session setup path that has an active name translation facility until the origin LU alias name is determined. Also, if VTAM knows the destination LU name is real, it still invokes the alias name translation facility to try to translate the COS and logon mode names and to determine the owning SSCP.
- If you specify the owning SSCP in the alias tables, use the name specified on the SSCPNAME start option used by the VTAM that owns the resource.
- The logon mode name, class-of-service name, and the origin LU name can be translated at the gateway VTAM that translates the destination LU name or any subsequent gateway or gateway-capable SSCP along the session setup path.

Prior to VTAM V3R2, the class-of-service name for the destination network had to be translated at the same gateway VTAM that translates the destination LU name. If the destination LU were a real name, determined by CDRSC definition, the COS name had to be translated at the same host.

- NetView domain IDs (NCCFID) must be unique and cannot be assigned alias names.
- Alias name translation facility support for LU 6.2 is restricted. This facility can be used to determine the correct network identifier of the destination logical unit and the name of the owning SSCP. However, it cannot be used to translate logical unit names. Most peripheral nodes that use LU 6.2 communication protocol do not support alias name translation.
- The VTAM constants module limits the information to be translated by the alias name translation facility.

Resolving Alias Name Requirements

An alias name is defined in a host to represent a logical unit, logon mode table, or class-of-service (COS) name in another network.

The following method can be used to determine which alias names are required and to distribute the required alias name definitions to the correct gateway SSCPs. The alias names you choose cannot match the names of any resources in your network.

1. Create a table listing all of your LUs, even if they do not have cross-network sessions. You need not list every resource in your network if it is obvious from your naming conventions that certain names cannot be duplicated in any network to which you are connecting.
2. List all of the resources in other networks with which you communicate. If you use an adjacent SSCP list and do not specify the destination network, include all resources in all networks in the second column, even those with which you have no sessions. (This includes resources such as lines and physical units.) Otherwise, a resource might receive a session request that is intended for an LU that has the same name but is in a different network.

Table 28 on page 489 illustrates this process:

All Resources in Your Network (NETA)	All Other-Network Resources That Communicate with Owning Network	
ABLE	BAKER	NETY
CHARLIE	BAKER	NETZ
DOG	CHARLIE	NETX
FOX	EASY	NETX
EASY	JOAN	NETY
MARY	PETER	NETZ
ROGER	.	.
.	.	.
.	.	.
.	.	.

Table 28. Resources That You Communicate with in Other Networks

3. Find any duplicates in the table you have created.

In the example shown in Table 28, the duplicate names are CHARLIE and EASY. NETX.CHARLIE and NETX.EASY need alias names so you can distinguish them from resources in your network. Either NETY.BAKER or NETZ.BAKER needs an alias name so that you can distinguish them from each other.

For this example, the alias name definitions that you need to code for the NetView alias name translation facility are the following:

```
ORIGNET  NETA
          LU  CHARLIE,NETX,ALCHARLI
          LU  EASY,NETX,ALEASY
          LU  BAKER,NETY,ALBAKER
```

ORIGNET always represents the network that knows the resource by its alias name. The network specified on the LU definition statement is always the network that knows the resource by its real name.

With these alias names replacing the real names, you now know every resource you communicate with by a unique name.

After every network goes through the preceding process, consolidate the results so that every network that is to request name translations has a complete set of alias name definitions for LUs. For example, if NETX has a NetView alias name translation facility, you should supply NETX with the following:

```
ORIGNET  NETA
          LU  CHARLIE,NETX,ALCHARLI
          LU  EASY,NETX,ALEASY
```

Similarly, if NETY has a NetView alias name translation facility, you should supply NETY with the following:

```
ORIGNET  NETA
          LU  BAKER,NETY,ALBAKER
```

If your network has a NetView alias name translation facility, you should keep a copy of all the definitions you created. In addition, other networks might give you alias name definitions that they created to refer to your resources. For example, NETX might give you the following:

```
ORIGNET NETX
  LU MARY,NETA,MARYX
  LU EASY,NETA,EASYX
  LU CHARLIE,NETA,CHARLIE
```

By distributing the alias name definitions as indicated, you might find that several networks have the same definition. However, you are assured that LU names get translated at the first NetView alias name translation facility called, regardless of the direction of session setup.

Establishing and Controlling SNA Sessions

This section contains information to be considered for non-native network type 2.1 connections. Also, to control cross-network sessions you can use the automatic logon function of VTAM, and **MVS, VM** take advantage of certain NetView Performance Monitor functions.

Non-Native Network Type 2.1 Connections

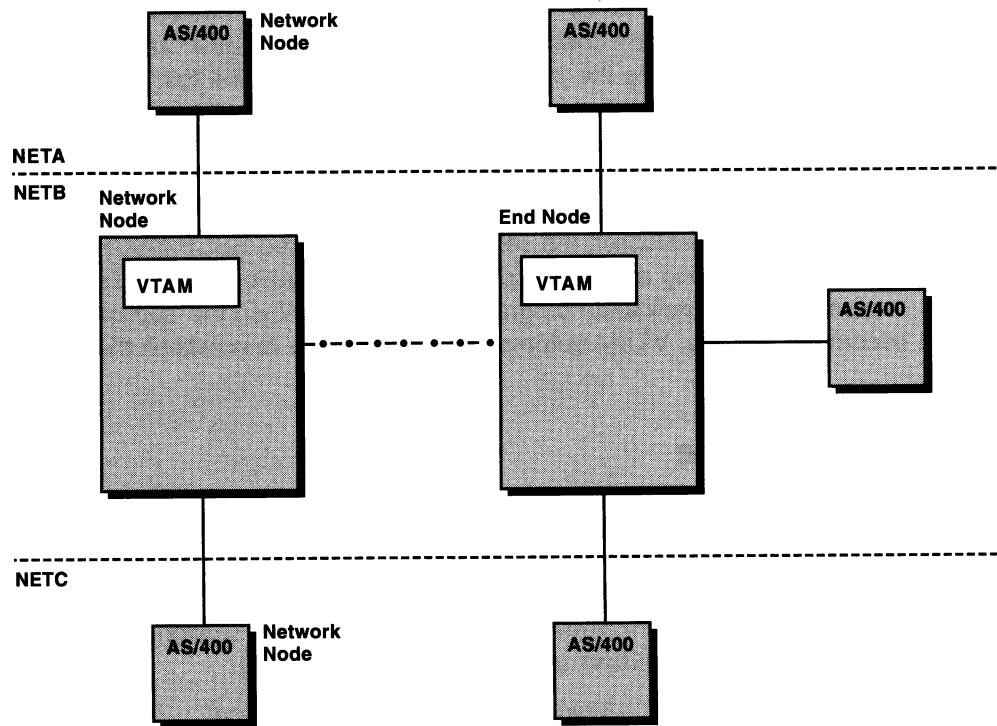
For non-native network type 2.1 connections, consider the following when defining and specifying destination logical units.

Considerations for Network Nodes

In Figure 121 on page 491, two AS/400 network nodes are attached to a VTAM network node by non-native network type 2.1 connections. If two resources (one at each of the AS/400 network nodes) are to communicate with each other through the intermediate VTAM network node, at least one of the connections must be a LEN connection or APPN multiple network connectivity support must be implemented (see “APPN Multiple Network Connectivity” on page 545).

Considerations for End Nodes

In Figure 121 on page 491, some resources are attached to the end node by non-native network type 2.1 connections. If one of the resources initiates a session, the network node server caches the network-qualified name of the originating LU, and the resource can be found subsequently by the APPN network. However, APPN may not be able to locate these resources when attempting to locate them as destination LUs, unless the resources are predefined at the end node (with CDRSC definitions for all possible network IDs) and registered with the network node server. If all possible network-qualified names of the resource are known at the network node server, APPN searches can locate the destination LU. The use of a predefined alias name for these resources at the end node is not recommended.



APPN Connection -.-

Figure 121. Non-Native Network Type 2.1 Connections

Specifying and Defining the Destination LU

When requesting a session with resources that are attached by non-native network type 2.1 connections, the originating LU should specify the NETID of the adjacent node or no NETID. If the originating LU specifies the NETID of the adjacent node, the destination LU must be located in the same network as the adjacent node. If a session request does not include a NETID for the destination LU and a predefined alias CDRSC is defined for the destination LU, then the resource can be located in any network.

Automatic Logon

Using the LOGAPPL operand, you can define a cross-network controlling primary logical unit on any peripheral node (local or remote, switched or nonswitched). The effect is essentially the same as if a CDRSC within the same network were the controlling primary logical unit, except that the resulting session is cross-network rather than within the same network. The network-qualified name of the controlling primary logical unit can be specified with the LOGAPPL operand.

In a single-network environment, when the SSCP-SSCP session necessary for the establishment of a controlling primary logical unit session is not immediately available, VTAM stores information necessary for session setup. When the necessary SSCP-SSCP session becomes active, VTAM establishes the LU-LU session.

However, this process does not occur for cross-network sessions. You can establish cross-network controlling primary logical unit sessions through any number of intermediate SSCPs only if all of the required SSCP-SSCP sessions are active when the LU-LU session setup is attempted. In the case where the session setup

path consists of more than one SSCP-SSCP session, the session setup proceeds as far as establishing the session control blocks at each active SSCP along the setup path. At some point along the path, an SSCP can determine that the session request cannot be routed any further because it has no active sessions with any SSCPs in its adjacent SSCP list for the specific destination logical unit. If this occurs, the request is rejected, the session control blocks are freed, and the operator is notified of the failure. After the necessary SSCP-SSCP sessions are active, the operator must try to re-establish the session. If all required SSCP-SSCP sessions exist, but the controlling application program is unavailable, VTAM fails the initiation request until the application program is available. When the application program becomes available, VTAM automatically attempts to re-establish the session.

Session Management Exit Routine

VTAM calls the session management exit routine whenever a session between logical units is being established, terminated, taken over, or recovered. The session management exit routine is invoked by every VTAM in the session-setup path. The session management exit routine can be invoked to perform the functions discussed in “Session Management Exit Routine” in *VTAM Customization*. Some functions that you can perform in this exit are the following:

- Check or restrict the use of a logical unit (session authorization).
- Gather accounting information (session accounting).
- Select adjacent SSCP for LU-LU session and resource status requests.
- Process a session takeover by the alternate XRF application program.

Code a session management exit routine if you want to use these functions. There is no default session management exit routine.

Session Accounting

With the capability of peripheral LUs serving as primary LUs (PLUs), accounting application programs relying on the primary LU to be located only in the host might not be sufficient. NCP collects byte and PIU counts associated with each half-session component of a session and reports that information, along with session awareness data, to NetView Performance Monitor (NPM).

Especially important to a multiple-network environment is the ability of the user to specify the following during NCP generation:

- Whether gateway session accounting is to be used for cross-network sessions
- The number of half-sessions for which session accounting is done.

At LU-LU session setup time, a session awareness PIU is sent to NPM as unsolicited information. Byte and PIU counts are updated every time there is traffic on the session. When the counts surpass the user-specified threshold, the counters are sent to NPM. At the end of the session, session awareness data and the last set of counters for the session are sent to NPM. The user-specified options as well as thresholds can be changed using NPM commands. In addition, transport of accounting data to NPM can be suspended and resumed using NPM commands.

NPM provides gateway session accounting for cross-network sessions.

Operating VTAM

This section contains information about operation considerations in a multiple-network environment. If you are migrating to a multiple-network, take this information into consideration when making decisions about your operating procedures.

Activating Resources

Before logical units in one network can establish sessions with logical units in another network, successive sessions between gateway SSCPs connecting the networks must be established.

For an SSCP-SSCP session, if an NCP on the originating side of a gateway is not available, a session activation request remains pending. If an NCP is not available on the destination side, a session activation request fails.

In interconnected networks, the operator can test routes starting in a gateway NCP by using the NETID operand of the command with the ORIGIN operand.

Gateway NCP and VTAM interaction

A gateway VTAM does not have to reside in the gateway NCP's native network, but having the gateway VTAM in the NCP's non-native network puts some restrictions on VTAM. Only a small number of commands from a VTAM in a non-native network are accepted by the gateway NCP: activating, deactivating the gateway NCP, route testing, and requests for address translations. Only VTAMs in the native network can request dynamic dumps or traces of resources.

You can prevent link-attached hosts in the non-native network from loading or dumping the gateway NCP by specifying IPL=NO on the LINE definition statement for that link. You can prevent VTAMs in the non-native network from activating the gateway NCP by specifying ACTPU=NO on the NETWORK definition statement for that network.

Terminating Sessions

When any SSCP-SSCP session along a cross-network session path of an LU-LU session is disrupted, the only termination requests that can be handled are those that are forced, such as VARY INACT with the FORCE operand or VARY TERM with the FORCE operand.

Using the NetView Program for Network Management

The NetView program provides facilities for problem determination and network management in a gateway configuration. These facilities include the following:

- Information about the alias addresses, alias and real LU names, and owning SSCPs for LUs involved in cross-network sessions
- Session awareness data, such as session protocols
- Inquiry, on a session basis, as to whether a complete physical path exists between the controlling subarea nodes of the two session partners
- Display of the subarea nodes and transmission groups.

The availability of these functions for a particular cross-network session depends on the NetView program being installed in the appropriate hosts. It also depends on the appropriate level of VTAM or NCP being installed in each subarea.

It is recommended that the NetView program be installed at every gateway VTAM host and at every host that controls resources that might have cross-network sessions. This ensures that the NetView program is able to provide the maximum amount of problem determination data for cross-network sessions.

Verifying a Multiple-Network Environment

This section suggests steps to follow to verify that your multiple-network environment functions properly. However, before you perform the steps in this section, perform the steps in both “Verifying a VTAM Network” on page 160 and “Verifying a Multiple-Domain Subarea Network” on page 435 (in that order) for each of your networks.

For specific information about how to perform these steps and for the syntax of operator commands, refer to *VTAM Operation*. If these steps do not yield the expected results, double-check the definition statements for the resources that are affected.

To verify that your multiple-network environment functions properly, do the following:

1. Activate each gateway VTAM and gateway NCP.
2. Activate major nodes required for cross-network sessions.
3. Establish required cross-network SSCP-SSCP sessions.
4. Verify cross-network sessions by exchanging requests across networks, between terminals and hosts, and between application programs.
5. Verify termination processing when an SSCP-SSCP or SSCP-gateway NCP session is deactivated.
6. Try backup and recovery procedures.
7. Verify the NetView program, if it is installed.
8. Halt VTAM in each domain in each network.

Part 4. Implementing APPN in Your Network

Chapter 9. Implementing a Combined APPN and Subarea Network	497
Implementing a Combined APPN and Subarea Network	497
Starting VTAM	501
SORDER Start Option	501
SSEARCH Start Option	502
CDRSCTI Start Option	502
Dependent LUs	502
Dependent LU Server	502
Dependent LU Server Support Across Subnetwork Boundaries	506
Defining CDRSCs	508
Shared NCPs	508
SSCP Takeover	509
Establishing and Controlling Sessions	512
Controlling Searches	512
Using the SORDER Start Option to Control Network Search Order	512
Using the SSEARCH Start Option to Limit Subarea Network Searches	513
Using the CDRSCTI Start Option to Reduce Broadcast Searches of APPN	513
Using the DISJOINT Operand to Define Disjoint SNI Networks	513
Virtual-Route-Based Transmission Groups (VR-Based TGs)	515
APPN and Subarea Class of Service Resolution	520
Adding and Moving Nodes	522
Verifying a Combined APPN and Subarea Network	523
 Chapter 10. Implementing an APPN Network	 525
Implementing an APPN Network	525
Starting VTAM with APPN Support	525
Enabling Control Sessions	525
CP-CP Sessions Between Two VTAM Nodes	526
Connecting an APPN Node to the Network	527
Connections through Boundary Function-Based Transmission Groups	527
Multiple Connections with Parallel Transmission Groups	528
Connection Network	528
Defining an APPN Connection	533
Defining a Network Node Server List	544
APPN Multiple Network Connectivity	545
Peripheral Subnetwork Boundaries	548
Extended Subnetwork Boundaries	548
Using APPN Multiple Network Connectivity Support	549
Network Routing and Resource Location for APPN Nodes	550
Types of Searches	551
Minimizing Broadcast Searches to Improve Performance	553
Checkpointing of the TRS Database and the Directory Database	555
APPN Class of Service	556
Verifying a VTAM APPN Network	560

Chapter 9. Implementing a Combined APPN and Subarea Network

Use this chapter to implement a network combining both APPN and subarea nodes and links.

This chapter builds upon previous chapters to avoid repetition. Therefore, you should be familiar with Chapter 5, "Getting Started with VTAM - Overview" on page 125, Chapter 6, "Getting Started with VTAM - Detail" on page 161, and Chapter 7, "Implementing a Subarea Network" on page 389 before reading this chapter.

Chapter 10, "Implementing an APPN Network" on page 525 provides further information on implementing APPN in your network.

Implementing a Combined APPN and Subarea Network

A sample communication management configuration (CMC) is shown in Figure 122 on page 498. This configuration is not intended to be typical or recommended, but is used here to demonstrate converting to APPN.

Before using the APPN capabilities of VTAM, any communication controller that is to support APPN links (all of them in Figure 122 on page 498) must be a 3745 with NCP Version 6 Release 2 or later release. Communication controllers that do not support APPN links do not require NCP Version 6 Release 2 or later.

Use the following steps to begin to convert this configuration to APPN:

1. Determine which connections to type 2.1 nodes that you want to be LEN connections, and which connections you want to be APPN connections.

All connections between the NCPs and type 2.1 nodes are automatically tried as APPN connections when APPN support is specified, unless you explicitly designate them as LEN connections using the CONNTYPE=LEN start option or the CONNTYPE=LEN operand on the GROUP, LINE, or PU definition statement. By specifying CONNTYPE=LEN initially, connections can later be converted to APPN in phases.

Notes:

- a. Prior versions of VTAM did not enforce CPNAME uniqueness for LEN nodes, except when CPNAME is coded on a switched PU definition statement. It is possible that your network has PUs with duplicate CPNAMEs. This CPNAME duplication should be resolved prior to a node becoming an APPN node. Specifying CONNTYPE=LEN for a connection causes VTAM to avoid checking for duplicate CPNAMEs, except when CPNAME is coded on a switched PU definition statement.
- b. When converting a connection from LEN to APPN, ensure that the adjacent link station name (VTAM-defined PU name) for an adjacent node is unique from the CP name of that node.

If you want most or all connections to type 2.1 nodes to initially be LEN connections, code the CONNTYPE=LEN start option at the CMC host. Then, if

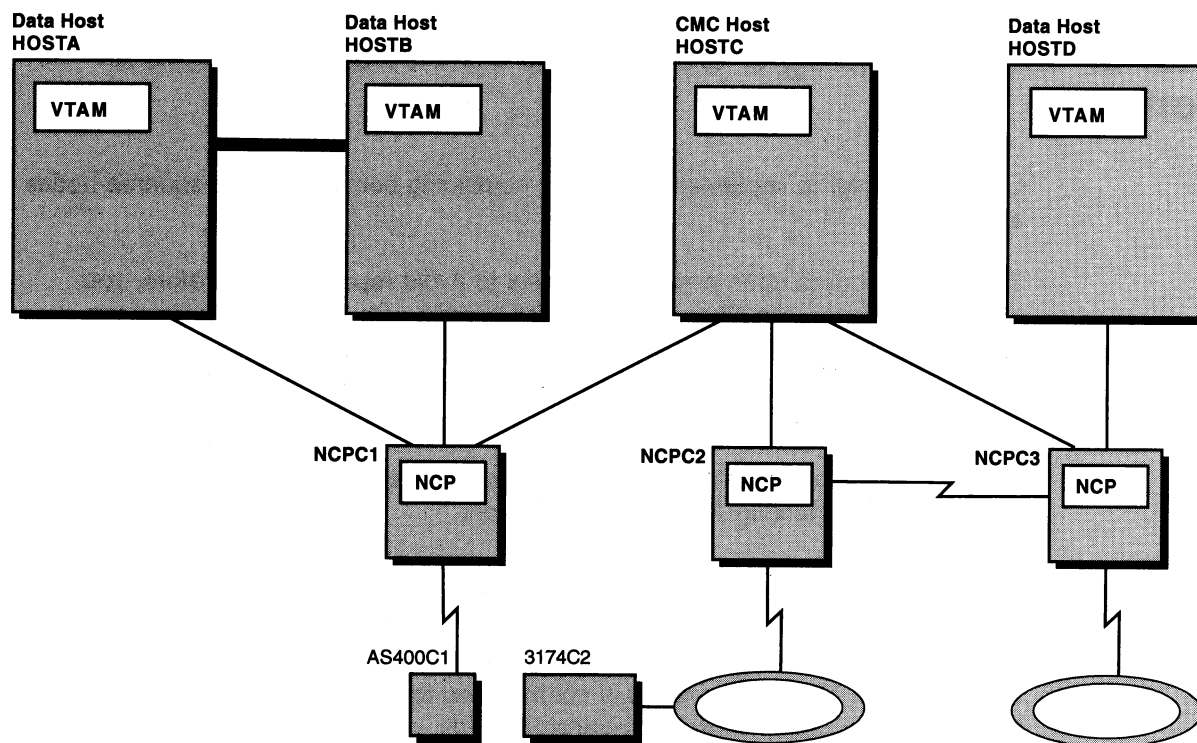


Figure 122. Example of Communication Management Configuration

you do not code CONNTYPE=APPN on any GROUP, LINE, or PU definition statement, all connections to type 2.1 nodes are LEN connections.

If you want only particular connections to type 2.1 nodes to initially be LEN connections, do not use the CONNTYPE=LEN start option. Instead, code CONNTYPE=LEN on the appropriate GROUP, LINE, or PU definition statements.

2. Code the NODETYPE=NN start option at the CMC host.

The NODETYPE start option is required to specify APPN support. Assuming that the HOSTSA start option is also coded for HOSTC, coding NODETYPE=NN makes this VTAM an interchange network node. An interchange network node supports subarea and APPN protocols, and is capable of transforming subarea protocols to APPN protocols, and vice versa.

To APPN, the CMC host and all owned NCPs (NCPC1, NCPC2, and NCPC3) become one composite network node as shown in Figure 123 on page 501. At this point, however, provided that all connections to type 2.1 nodes were specified as LEN connections in step 1 on page 497 using CONNTYPE=LEN, VTAM and NCP perform only their traditional subarea functions. If VTAM is started in HOSTC, no CP-CP sessions would be started between the composite network node and the adjacent nodes because connections to type 2.1 nodes are LEN connections and the adjacent VTAMs do not have APPN support.

3. Depending on your conversion strategy determined in step 1, convert LEN connections to APPN connections as desired by:

- Changing CONNTYPE=LEN to CONNTYPE=APPN (or simply deleting CONNTYPE=LEN) on the appropriate GROUP, LINE, or PU definition

statements (and inactivating and reactivating the appropriate links), or restarting VTAM with the CONNTYPE=APPN start option.

- Specifying any other desired APPN operands on the GROUP, LINE, or PU definition statement (for example, NN, CPCP or TGP).

Note: If the NN operand is not specified, the node type (NN or EN) of the adjacent type 2.1 node is determined when the connection is activated. If specified incorrectly, the connection will fail.

- Defining the connection as APPN at the adjacent type 2.1 node.

In the example in Figure 123 on page 501, all LEN connections are converted to APPN connections. With APPN connections, HOSTC can be used as the network node server, and independent LUs can now set up LU-LU sessions with the composite network node using APPN searches and flows. In addition, if a new independent LU is added to a type 2.1 node that is adjacent to the composite network node, the independent LU must now be defined only to that adjacent type 2.1 node. VTAM requires no definition for the added LU.

4. Code the NODETYPE=NN start option at data host B. Code a local SNA major node (VBUILD TYPE=LOCAL) that defines the type 2.1 channel attachment between data host B and NCPC1. Code PUTYPE=2 on the PU definition statement in the local SNA major node.

Assuming that the HOSTSA start option is also coded at HOSTB, this VTAM is now an interchange network node as shown in Figure 123 on page 501.

To the composite network node, HOSTB appears as a network node. The connection between HOSTB and NCPC1 is now a type 2.1 connection. A CP-CP session is now possible between HOSTB and the composite network node. Thus, LUs connected to the composite network node through one of the NCPs can now set up LU-LU sessions with any application in HOSTB using APPN searches and flows. New applications added in HOSTB need not be defined to the composite network node.

Note that the change of the link between HOSTB and NCPC1 to an APPN link results in the creation of additional NCP control blocks, which could require additional storage space. Whether you see a net decrease or increase in NCP storage is dependent upon the decrease in path table sizes in relation to any increase in storage due to the number of cross-domain sessions that are now boundary function sessions over APPN connections.

5. Code the NODETYPE=EN start option at data host A. Again, use a local SNA major node (VBUILD TYPE=LOCAL) to define the type 2.1 channel attachment between data host A and NCPC1, and code PUTYPE=2 on the PU definition statement in the local SNA major node.

Assuming that the HOSTSA start option is also coded at HOSTA, this VTAM is now a migration data host as shown in Figure 123 on page 501. A migration data host node supports subarea and APPN protocols, but does not act as an intermediate node. A migration data host cannot activate any NCPs.

To the composite network node, HOSTA appears as an end node. The connection between HOSTA and NCPC1 is now a type 2.1 connection. A CP-CP session is now possible between HOSTA and the composite network node. Thus, LUs connected to the composite network node through one of the NCPs can now set up LU-LU sessions with any application in HOSTA using APPN searches and flows. New applications added in HOSTA need not be defined to the composite network node.

Again, note that the change of the link between HOSTA and NCPC1 to an APPN link results in the creation of additional NCP control blocks, which could require additional storage space.

A migration data host can also still participate in SSCP-SSCP sessions with another VTAM (for example, HOSTB). If the channel-to-channel (CTC) connection between HOSTA and HOSTB is not also converted to APPN, you can continue to use subarea CTC support over that connection.

6. In this example, the NODETYPE start option is not coded at HOSTD. HOSTD can still communicate with resources in the HOSTC domain using the subarea connection to NCP3. HOSTD can also still communicate with resources in the HOSTA and HOSTB domains using the interchange node function of HOSTC.

As a result of the above conversion steps, the following definitions are no longer required. You can remove them, but they do not have to be removed.

- All PATH definitions for routes between HOSTA or HOSTB and other subareas, except PATHs for the CTC connection between HOSTA and HOSTB.
- In HOSTA and HOSTB, the CDRM definitions for HOSTC and HOSTD
- In HOSTC and HOSTD, the CDRM definitions for HOSTA and HOSTB
- CDRSC definitions in HOSTA and HOSTB for LUs in HOSTC and HOSTD, if present
- CDRSC definitions in HOSTC and HOSTD for LUs in HOSTA and HOSTB, if present
- Wildcard definitions in 3174C2 and AS400C1 that allow requests for unknown resources to be forwarded to VTAM over LEN links (assuming the links to 3174C2 and AS400C1 are now APPN links)
- ALS selection logic in the session management exit that allows requests to be forwarded from VTAM to independent LUs over LEN links that have been converted to APPN
- APPN-capable PUs included on the ALSLIST operand on the CDRSC definition statement (they can be optionally replaced by a single ISTAPNPU entry, a reserved keyword that represents any APPN capable PU).

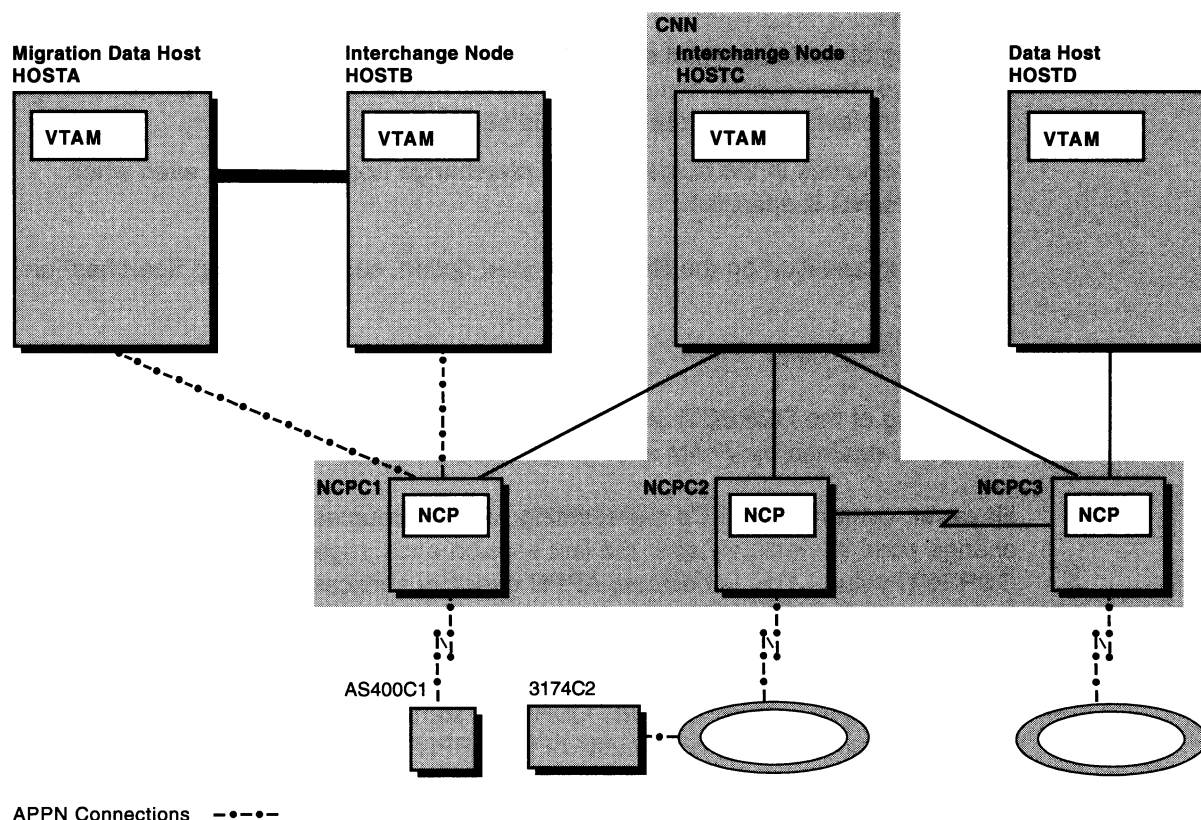


Figure 123. Communication Management Configuration After Conversion

Starting VTAM

In addition to the CONNTYPE, NODETYPE, and HOSTSA start options, some other start options used in a combined subarea and APPN environment are associated with locating resources. All new and changed start options default to optimal values for an APPN network. During conversion, however, these defaults can be modified based on your particular conversion strategy and configuration.

SORDER Start Option

The setting of the SORDER start option controls the order in which the APPN and subarea networks are searched when a search request arrives at an interchange node. By default, the APPN network is searched first. This start option is overridden if the target LU's location is already known and SSCORD=PRIORITY is specified. This start option can also apply to migration data hosts.

For further information on the SORDER start option, see "Controlling Searches" on page 512.

SSEARCH Start Option

The setting of the SSEARCH start option determines whether the subarea network is searched when search requests from the APPN network arrive at an interchange node. By default, the subarea network is searched.

Note: Resources in the domain of the interchange node are found even when SSEARCH=NO is specified.

For more information on the SSEARCH start option, see “Controlling Searches” on page 512.

CDRSCTI Start Option

The setting of the CDRSCTI start option controls the amount of time that a dynamic CDRSC is retained by VTAM after its last session ends.

However, dynamic CDRSCs representing APPN resources are freed by an interchange node immediately after the last session ends, regardless of the specified CDRSCTI value. This is because APPN directory services in the interchange node has an entry for each APPN resource.

Dynamic CDRSCs representing subarea resources are retained for eight minutes (by default). You can increase the CDRSCTI start option value to minimize the potential for unnecessary broadcasts of the APPN network for subarea resources.

For further information on the CDRSCTI start option, see “Controlling Searches” on page 512.

Dependent LUs

VTAM extends support for dependent (non-LU 6.2) LUs to APPN networks and combined subarea and APPN networks. Dependent LUs are owned by VTAM and rely on the SSCP to start sessions. There are SSCP-PU and SSCP-LU sessions for dependent LUs. Dependent LUs can start a session when the primary LU supports SLU-initiated sessions (for example, as does a VTAM application program). Dependent LUs can be the secondary LU in sessions with VTAM application programs in APPN, subarea, and combined APPN and subarea networks.

Note: If bisynchronous (BSC) 3270 sessions are required, traditional subarea paths must be retained for routing these sessions. These same subarea paths can be shared by APPN sessions using VR-based TG, permitting coexistence of APPN and BSC sessions.

Dependent LU Server

The dependent LU server (DLUS) function facilitates conversion from a subarea environment to an APPN environment, allowing you to maintain central management of remote dependent LUs while benefiting from APPN throughout a network.

Two LU 6.2 sessions (one inbound, one outbound) are established between a DLUS and a dependent LU requester (DLUR), and these LU 6.2 sessions are collectively known as the CPSVRMGR pipe. SSCP-PU and SSCP-LU session flows use the CPSVRMGR pipe. An SSCP-PU session is established between a VTAM network node and the dependent LU's owning PU, and an SSCP-LU session is established between VTAM and the dependent LU. Session initiation flows for the

dependent LU are sent over the SSCP-LU session, and VTAM can use subarea or APPN flows to initiate a session with the PLU. BIND and session data are then routed directly between the PLU and the dependent LU.

For example, in Figure 124, VTAMB is a dependent LU server for LU4 and LU5. SSCP-PU sessions are established between VTAMB and PU4 and between VTAMB and PU5, and SSCP-LU sessions are established between VTAMB and the dependent LUs (LU4 and LU5). Session initiation flows for the dependent LUs are sent over the SSCP-LU sessions. VTAMB uses APPN flows to initiate a session with a PLU that VTAM can communicate with through an APPN network (for example, LU2 and LU3), and subarea flows to initiate a session with a PLU reachable through a subarea network (for example, APPLA1). The BIND and session data are then routed directly between the PLU and the dependent LUs.

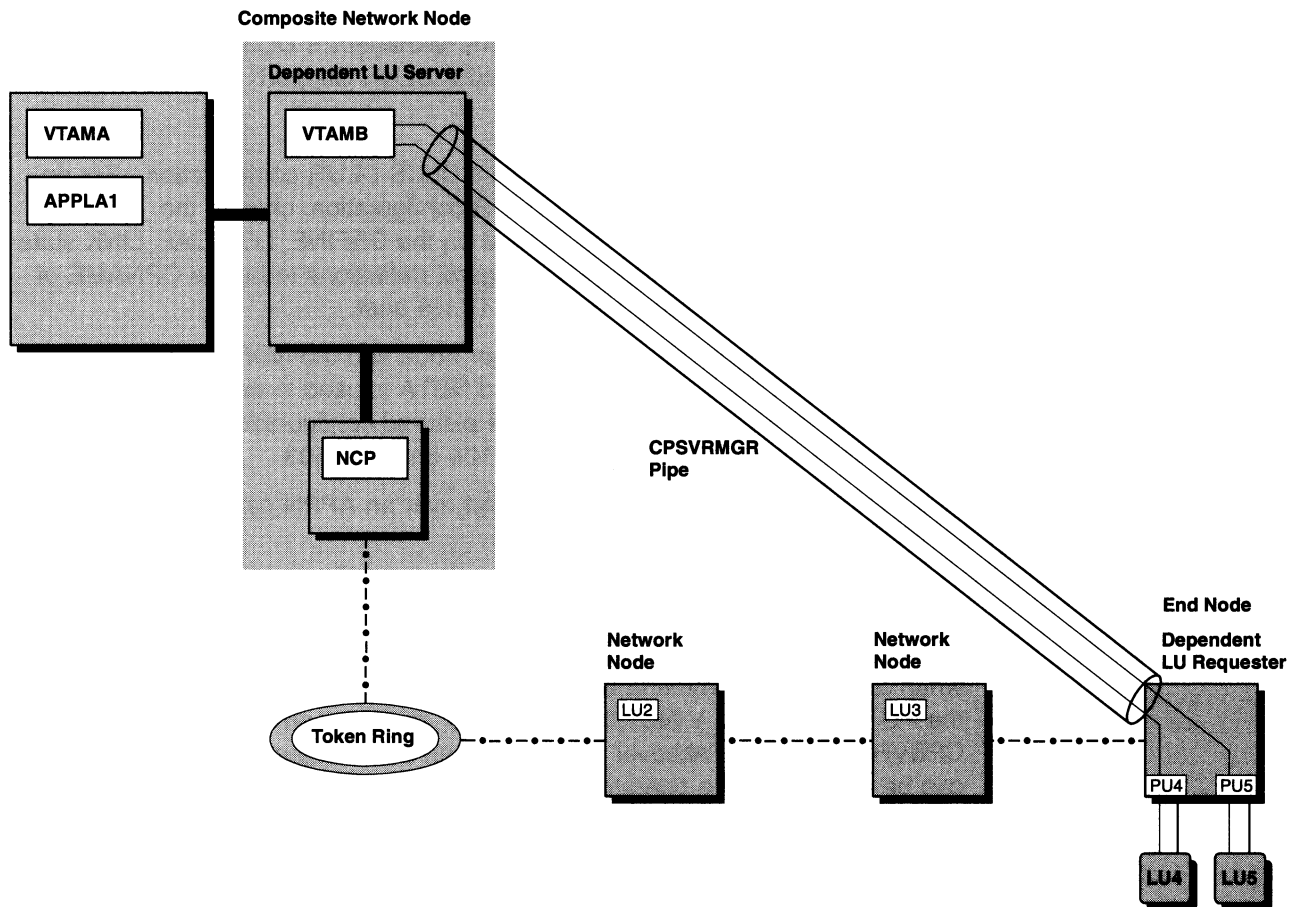


Figure 124. VTAM Functioning as a Dependent LU Server

To implement the dependent LU server function, consider the following:

- For DLUR-initiated PU activation, no system definition is required. The dynamic switched definition facility can be used to dynamically define the PU. For information on the dynamic switched definition facility, see “Defining Switched Resources Dynamically” on page 143 and “Dynamically Defining Switched Resources” on page 260.
- Dynamically defined PUs can be displayed using the DISPLAY ID command, specifying the DLUR CP name on the ID operand. This shows all PUs that currently have an SSCP-PU session (both dynamic and predefined PUs). You

can also determine which LUs currently have an SSCP-LU session by displaying a specific PU.

- For VTAM to initiate activation of a PU, define the dependent LU requester by including the DLURNAME and DLCADDR operands on the PATH definition statement in the switched major node (regardless of whether its connection to the DLUR is switched or leased). DLURNAME specifies the CP name of the DLUR that owns the PU, and DLCADDR includes data link control (DLC) information used by the DLUR to locate the PU. Also include the MAXDLUR operand on the VBUILD definition statement to indicate the maximum number of unique DLURs defined for this switched major node.

Note: If the DLURNAME is not fully qualified, the NETID of the DLUS is used.

A sample switched major node to match Figure 124 on page 503 is shown in Figure 125 on page 505. For a sample of the definitions required at the DLUR and other information on Communications Manager/2's DLUR support, see Appendix L, "Communications Manager/2's DLUR Support" on page 705.

Notes:

1. This is an example of a non-adjacent DLUS-DLUR configuration. For an example of an adjacent DLUS-DLUR configuration, change the FQ_ADJACENT_CP_NAME defined on the DEFINE_LOGICAL_LINK statement in the Communications Manager/2 definitions to be the CPNAME of the VTAM that is configured as the DLUS host.
 2. DLUR1 is used in the switched major node and the NDF and RSP files as the name of the pictured DLUR, and NETA is used in the NDF and RSP files as the network identifier for the pictured configuration, although they are not actually depicted in Figure 124 on page 503.
- The DLUS and DLUR must be connected over an APPN path that supports the CPSVRMGR class of service.
 - Takeover and giveback is supported for DLUR-attached PUs. The PU must be defined in VTAM with ANS=CONT, and the DLUR must be capable of handling ANS=CONT support. Giveback processing for DLUR-supported PUs occurs when an operator command is issued from the DLUS to relinquish control of the PU (for example, V NET,INACT,ID=dlur_pu,G), or an outage occurs for the CPSVRMGR pipe. Takeover processing can occur automatically for redial or can be initiated by the operator via a VARY DIAL command, for PUs previously released via giveback processing.

```

SAMP5WMN VBUILD TYPE=SWNET,MAXDLUR=4
PU4      PU  ADDR=01,                      X
          IDBLK=05D,                      X
          IDNUM=00001,                    X
          USSTAB=AUSSTAB,                 X
          MODETAB=AMODETAB,                X
          ISTATUS=ACTIVE,                  X
          MAXPATH=3
PATHU4   PATH PID=1,                      X
          DLURNAME=DLUR1,                  X
          DLCADDR=(1,C,INTPU),             X
          DLCADDR=(2,X,05D00001)
LU4      LU  LOCADDR=1,                    X
          PACING=(1,1),                    X
          VPACING=2
*
PU5      PU  ADDR=02,                      X
          IDBLK=05D,                      X
          IDNUM=00002,                    X
          USSTAB=AUSSTAB,                 X
          ISTATUS=ACTIVE,                  X
          MAXPATH=1
PATHU5   PATH PID=1,                      X
          DLURNAME=DLUR1,                  X
          DLCADDR=(1,C,INTPU),             X
          DLCADDR=(2,X,05D00002)
LU5      LU  LOCADDR=1,                    X
          PACING=(1,1),                    X
          VPACING=2,                      X
          MODETAB=AMODETAB

```

Figure 125. Switched Major Node for a Dependent LU Server

Notes:

1. If using NCP, NCP Version 6 Release 2 or greater is required.
2. The VARY INACT,TYPE=GIVEBACK command is used to release a DLUR-attached switched PU (and its subordinate LUs) from an SSCP's ownership. To utilize this function, the DLUR must support ANS=CONT.
3. A DLUS may serve multiple DLURs simultaneously.
4. A DLUR may be served by multiple DLUS VTAMs simultaneously.
5. Multiple DLURs may support the same PU (one at a time). The SSCP-PU session is through only one DLUR at any given time.
6. DLUS function cannot be used with XRF or cryptography in an XRF environment.
7. Redial occurs as follows for DLUS supported PUs:
 - When a DLUS initiates the activation of a PU but receives a negative response, VTAM attempts to redial over each valid path statement for the PU until successful or until all valid paths have been tried. Redial does not occur if the negative response indicates that the PU is already active, or that the fully-qualified PCID is not unique. If the fully-qualified PCID is not unique, the DLUS attempts to redial the PU over the same path with a newly generated PCID.
 - Redial does not occur if PU activation was initiated by the DLUR.
 - When a protocol violation, TDU error, or CPSVRMGR session outage signal is received for a particular DLUR, VTAM attempts to redial every active or pending-active PU served by that DLUR for which a valid PATH statement is found. If the PU is already active, then VTAM performs the following:

- If the PU was defined with ANS=CONT and the DLUR supports this function, giveback processing is performed prior to attempting redial.
 - If the the PU was defined with ANS=STOP or the DLUR does not support ANS=CONT, the PU is deactivated prior to attempting redial.
- 8. When a CP-CP session between a DLUS and DLUR fails, the CPSVRMGR session is inactivated, enabling reactivation of the CP-CP session.
- 9. If the DLUS is located in a different APPN subnetwork than that of the DLUR or the PLU, APPN multiple network connectivity support is required (see “Dependent LU Server Support Across Subnetwork Boundaries”). If CPSVRMGR pipe initiation is attempted across an APPN subnetwork boundary and the DLUR does not support this function, a protocol violation is detected and the pipe is terminated.
- 10. Information regarding DLUR-attached resources can be processed by the configuration services XID exit. This exit routine can be coded so that VTAM either processes or denies requests for contact from known switched devices, or so that VTAM processes or denies requests for PU activation from a DLUR. For details, see *VTAM Customization*.
- 11. The IDBLK/IDNUM specified on the PU definition statement must match the value specified on the DLCADDR operand (when DLCADDR information is provided for IDBLK/IDNUM) on the associated PATH statement.
- 12. The IDBLK/IDNUM specified in the switched major node must match the NODE_ID value specified in the Communications Manager/2 RSP or NDF file, on the associated DEFINE_DEPENDENT_LU_SERVER statement.
- 13. The LOCADDR coded on the LU definition statement in the switched major node must match the NAU_ADDRESS value, when defined for a specific DLUR-supported LU (defined in Communications Manager/2 RSP or NDF files).

Dependent LU Server Support Across Subnetwork Boundaries

To use dependent LU server support across subnetwork boundaries, APPN multiple network connectivity support is required. NCP Version 7 Release 1 is also required for DLUS support across subnetwork boundaries.

If a DLUS is located in a different APPN subnetwork than a DLUR, the subnetwork of the DLUS must enable APPN multiple network connectivity support, as shown in Figure 126 on page 507.

Note: The node on the border of the DLUR's subnetwork (in Figure 126 on page 507, the network node in subnetwork B) can be a network node, peripheral border node, or extended border node.

Also shown in Figure 126 on page 507, if a DLUS is located in a different APPN subnetwork than the primary LU, both the subnetwork of the DLUS and the subnetwork of the primary LU must enable APPN multiple network connectivity support.

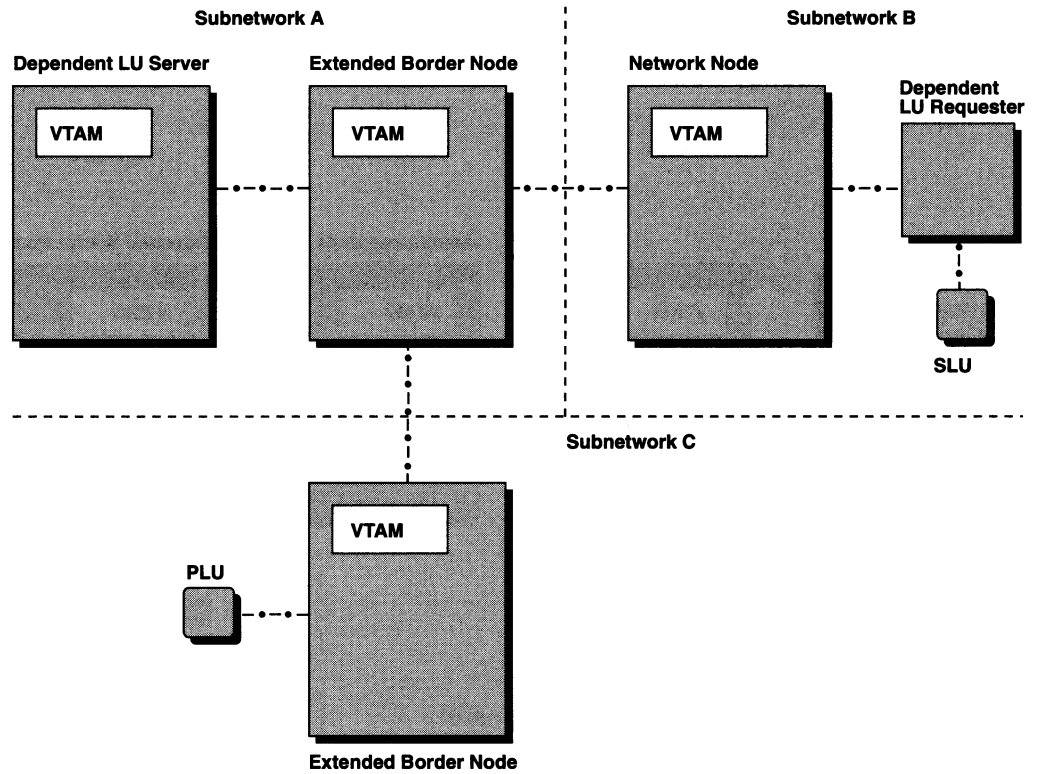


Figure 126. DLUS Located in Different APPN Subnetwork Than DLUR or PLU

If a primary LU exists in or through a subarea network, as shown in Figure 127 on page 508, all other APPN subnetworks through which the primary LU might be reached that use interchange nodes for connectivity to or through the subarea network must enable APPN multiple network connectivity support.

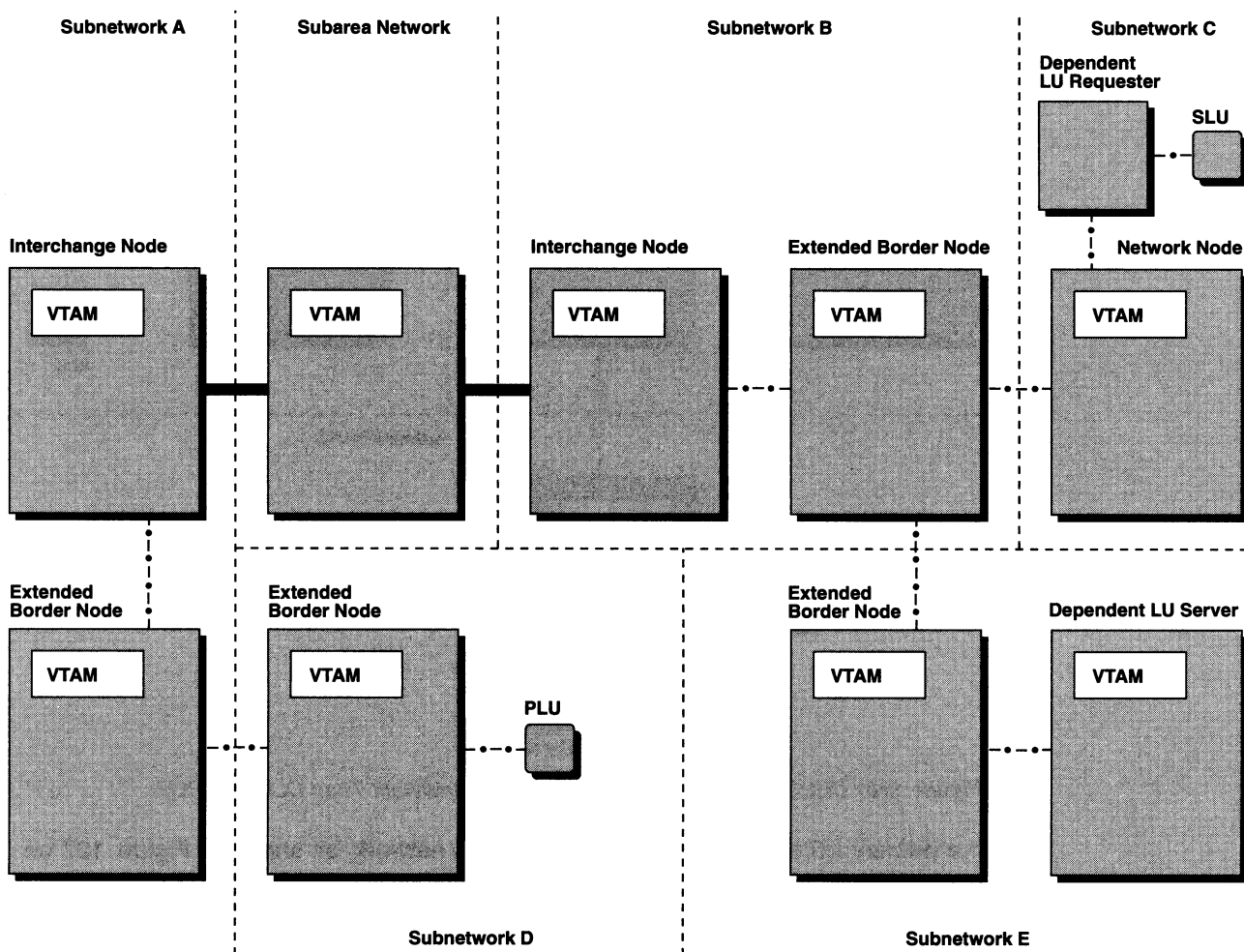


Figure 127. PLU Exists in or through a Subarea Network

For further information on APPN multiple network connectivity support, see “APPN Multiple Network Connectivity” on page 545.

Defining CDRSCs

When a resource is found in or through an APPN network, the interchange node through which the APPN network is entered becomes the owning CDRM of the resource to other VTAMs in the subarea network, unless the resource was predefined as a CDRSC with NETID and an owning CDRM name coded. If there are multiple interchange nodes, the owning CDRM can change, depending on the best search route at the time. If cross-domain resources (CDRSCs) are defined with the CDRM operand, they may have to be modified to specify an interchange node as the owning CDRM.

Shared NCPs

In a combined APPN and subarea network, VTAM provides for shared ownership of a local or remote NCP by two VTAMs.

If two VTAMs share an NCP, they both must have a subarea connection to that NCP and thus a subarea path between them. These two VTAMs can communicate

via an SSCP-SSCP session. The topology of the APPN network connected to one VTAM is not shared with the APPN network connected to the other VTAM, unless CP-CP sessions also exist between the two VTAMs. If CP-CP and SSCP-SSCP sessions exist between the two VTAMs, the decision on which control session to use for a specific LU-LU session setup depends on resource locations, start options, and tables.

SSCP Takeover

In a combined APPN and subarea network, VTAM provides full takeover capability.

A connection that is established as an APPN or LEN connection remains such until it is deactivated. However, an SSCP taking over an APPN connection views it as a LEN connection if the takeover SSCP has not implemented APPN. No CP-CP sessions can be established over the connection until the original SSCP regains control.

For example, in Figure 128, if HOSTA fails and HOSTB takes over, the connection from the NCP to the adjacent CP is viewed by HOSTB as a LEN connection if HOSTB has not implemented APPN. No CP-CP sessions can be established over the connection until HOSTA regains control.

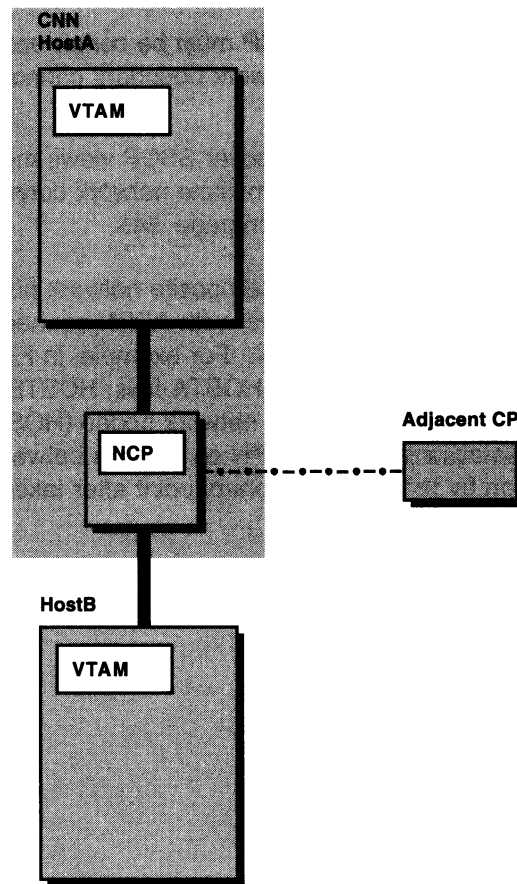


Figure 128. SSCP Takeover When Adjacent CP Does Not Support CP Name Change

If the takeover SSCP has implemented APPN, a connection is still treated as a LEN connection if both of the following are true:

- The takeover SSCP is not the original SSCP, in which case the adjacent CP has a connection to a different SSCP after takeover (CP name change)
- The adjacent CP does not support the changing of CP names.

Although the adjacent CP still views the connection as an APPN connection with the original SSCP, the takeover SSCP views the connection as an LEN connection. Therefore, no CP-CP sessions can be established over the connection until the original SSCP regains control.

For example, in Figure 128 on page 509, if HOSTA fails and HOSTB takes over, the CP name (from the adjacent CP's perspective) changes from HOSTA to HOSTB. If the adjacent CP does not support the changing of CP names, HOSTB treats the connection to the adjacent CP as a LEN connection (even if HOSTB has implemented APPN) and no CP-CP sessions can be established over the connection until HOSTA regains control. If the adjacent CP does support the changing of CP names, CP-CP sessions can be established when HOSTB takes over.

If the adjacent CP is located over a subnetwork boundary, the following conditions must also be met to allow a new CP-CP session with the takeover SSCP:

- The takeover SSCP must have APPN multiple network connectivity support.
- The definitions in the takeover SSCP must be consistent with those in the original SSCP (for example, a subnetwork boundary cannot be changed to a native connection during takeover).

If these conditions are not met, the takeover SSCP views the connection as a LEN connection. For information on APPN multiple network connectivity support, see "APPN Multiple Network Connectivity" on page 545.

If the adjacent control point is another composite network node as shown in Figure 129 on page 511, SSCP takeover with APPN connectivity occurs if the takeover SSCP has implemented APPN. For example, in Figure 129 on page 511, if HOSTB has implemented APPN and HOSTA fails, HOSTB can take over and a CP-CP session between the composite network nodes (HOSTC/NCP2 and HOSTB/NCP1) is established. The APPN connection between the NCPs is viewed as a LEN connection by the adjacent control point after takeover if the takeover SSCP has not implemented APPN.

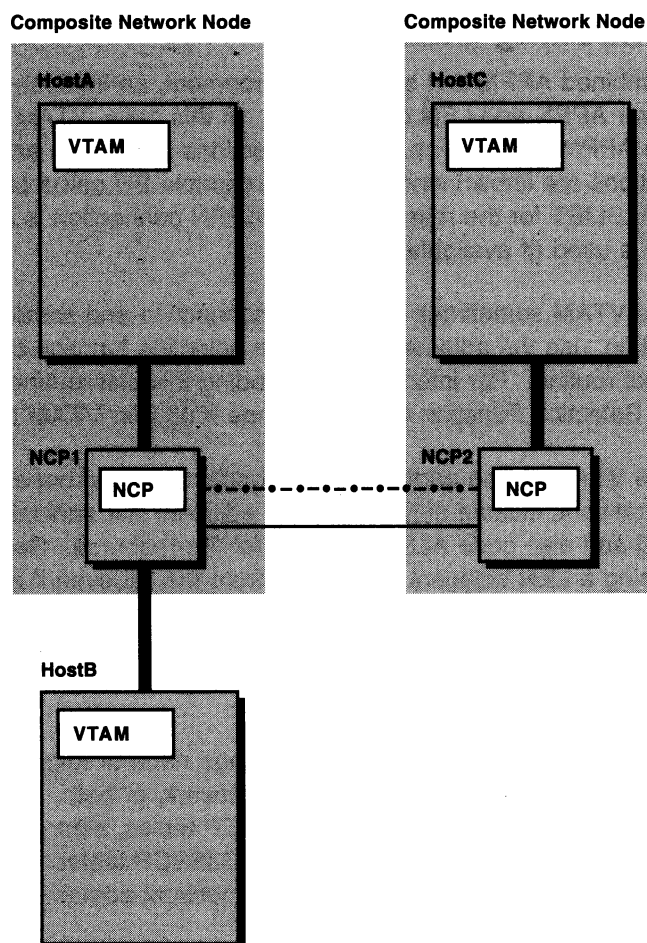


Figure 129. SSCP Takeover When Adjacent CP Is Another Composite Network Node

Notes:

1. In Figure 129, if HOSTA takes over for HOSTC or HOSTC takes over for HOSTA, APPN and LEN connectivity is not available after takeover (assuming a CP-CP session originally existed between only the composite network nodes shown) since the takeover SSCP cannot have a connection with itself (that is, after takeover, the takeover SSCP and the NCPs now form a composite network node). APPN and LEN connectivity is not available until the failing host regains control or another host (for example, HOSTB) takes control of the NCP originally owned by the failing host.
2. In Figure 129, a subarea connection and an APPN connection exist between the NCPs. If both the subarea connection and the APPN connection are active concurrently, the route used for communication between the NCPs is unpredictable. To control which link is used, you can choose to activate only one connection. For SSCP takeover, however, the subarea connection must be activated if not already active.

Establishing and Controlling Sessions

In a combined APPN and subarea environment, an independent LU can be accessible over APPN and LEN connections. In this case, VTAM always gives preference to APPN connections, if the connections are known and available. APPN connections are known when an APPN-capable PU name or ISTAPNPU is included in the ALSLIST for the resource. If no APPN connection is available, a LEN connection is used (if available).

To have VTAM sometimes use a LEN connection and sometimes use an APPN connection, use the adjacent link station selection function of the session management exit routine. For information on coding this exit routine, see “Adjacent Link Station Selection Function (Function Code X'08)” in *VTAM Customization*.

To force VTAM to use particular LEN connections and not allow a search of the APPN network, include only those LEN adjacent link stations on the ALSLIST operand and also code ALSREQ=YES for the resource. Session setup will continue using a LEN adjacent link station from the list even if a route exists through the APPN network.

Controlling Searches

When trying to locate a DLU, an interchange node or migration data host can search the APPN network, the subarea network, or both. The order that VTAM searches is based on the defined ADJSSCP tables, start option values, and the origin of the search request. Using the ADJSSCP tables and start options, you can concentrate the search on the part of the network containing the most frequently sought resources.

For further information on searching an APPN network, see “Network Routing and Resource Location for APPN Nodes” on page 550.

For further information on searching a subarea network, review the appropriate sections of Chapter 7, “Implementing a Subarea Network” on page 389, beginning with “Starting VTAM” on page 391.

Using the SORDER Start Option to Control Network Search Order

By default, the APPN network is searched first (SORDER=APPN), followed by the subarea network. Use the SORDER=SUBAREA start option to have the subarea network searched first during search attempts originating in the subarea network or in the interchange node itself. The SORDER=SUBAREA start option can also be used at a migration data host to have the subarea network searched first for requests initiated in the migration data host. If a search request is received from the APPN network, SORDER is ignored and the APPN network is still searched first.

To search part of the subarea network first, the APPN network second, and then the rest of the subarea network, code an ADJCDRM entry in the ADJSSCP table with the reserved keyword ISTAPNCP, and use the SORDER=ADJSSCP start option. The ISTAPNCP entry refers to the entire APPN network, and its placement in the ADJSSCP table indicates where in the search order the APPN network should be searched.

Notes:

1. If SORDER=APPN or SORDER=SUBAREA is specified, the placement of ISTAPNCP within the ADJSSCP tables is ignored.
2. The SORDER start option can be overridden if the target LU's location is already known and SSCORD=PRIORITY is specified.
3. If SORDER=ADJSSCP is specified and ISTAPNCP is not coded in the ADJSSCP table, the APPN network is not searched, unless SSCORD=PRIORITY is specified and a previous request for this resource determined that it was located in the APPN network.
4. Learned or predefined CDRM owners are always placed at the very top of the routing table, even if SORDER=APPN was coded. Thus, you can override the search order for particular CDRSCs by coding an owning CDRM name on the CDRSC definition statement.

Using the SSEARCH Start Option to Limit Subarea Network Searches

The SSEARCH start option controls whether the subarea network is searched when a network search request is received from the APPN network. YES is the default. Use NO to specify that the subarea network is not searched when a network search request is received from the APPN network. CACHE is also a possible value for the SSEARCH start option. CACHE specifies that the subarea network is not searched unless it is known that the target LU is located in the subarea network. This knowledge can exist as a result of system definition or as a result of an earlier successful search.

Using the CDRSCTI Start Option to Reduce Broadcast Searches of APPN

The setting of the CDRSCTI start option controls the amount of time that a dynamic CDRSC is retained by VTAM after its last session ends. If a dynamic CDRSC has been freed by VTAM in a combined environment, search requests originating in the subarea network or in the domain of the interchange node itself can result in unnecessary broadcasts to locate a subarea resource in the APPN network.

Therefore, to reduce broadcast searches of an APPN network that does not have a central directory server, you can increase the CDRSCTI value from the default of 8 minutes to as much as 7 days at each interchange node. This will minimize the potential for unnecessary broadcasts of the APPN network, since dynamic CDRSCs are retained for a longer period of time after the last session ends and VTAM has more of an opportunity to take advantage of priority routing to locate the resource in the subarea network.

Using the DISJOINT Operand to Define Disjoint SNI Networks

The DISJOINT operand on the CDRM definition statement in the cross-domain resource manager major node can also be used to control session request searches. At an interchange node, DISJOINT indicates whether an adjacent CDRM is disjoint from other SSCPs with the same NETID as the adjacent CDRM, and the only connection between the disjoint SSCPs is through the APPN network.

For example, in Figure 130 on page 514, HOSTA and HOSTB both have the same network identifier (NETA), but their only connection to each other is through adjacent network NETB.

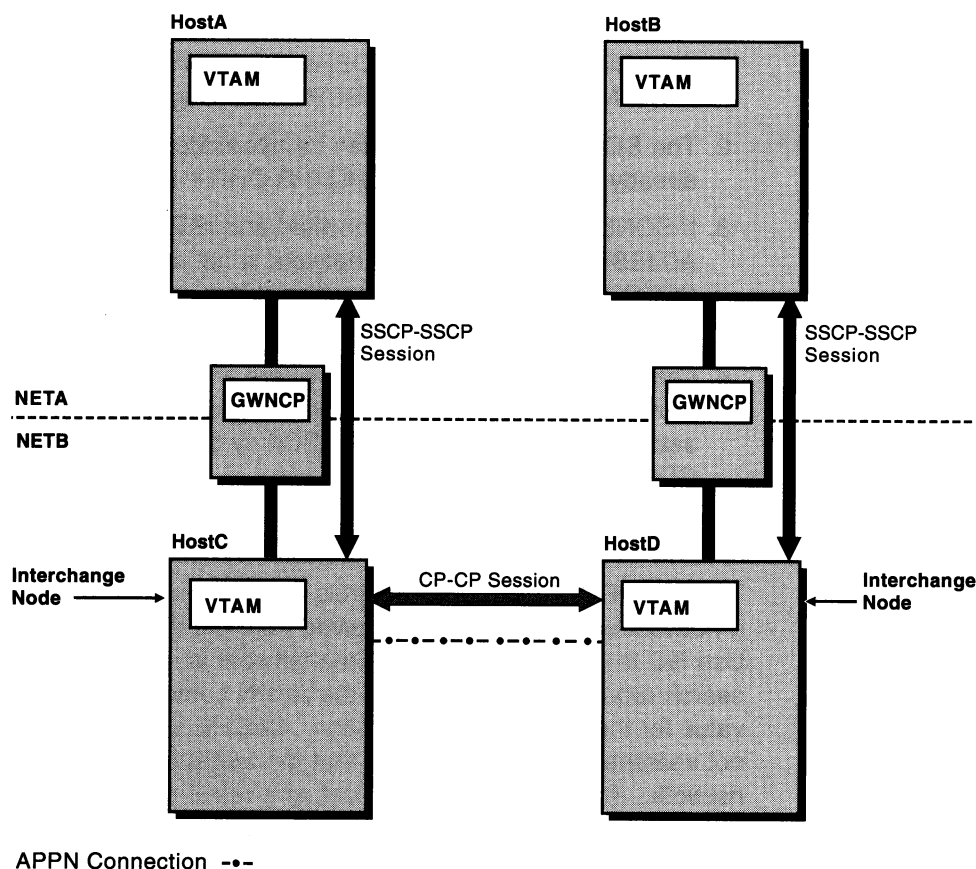


Figure 130. Disjoint Hosts

When a session request is received by HOSTC from HOSTA, HOSTC may decide to search the attached APPN network. When the search is forwarded into the APPN network, HOSTC includes information about the adjacent SSCP from which the request was received; namely, HOSTA's network identifier (NETA) and an indication of whether HOSTA is disjoint from other SSCPs within NETA (as defined by HOSTC using the DISJOINT operand in the CDRM definition statement for HOSTA).

When the APPN search request is received by HOSTD, HOSTD uses this information to determine the SSCPs to which the request should be routed. HOSTD always routes the request to SSCPs within HOSTD's network (NETB). HOSTD also routes the request to other SSCPs that are not in NETA (the network identifier provided by HOSTC, the interchange node serving as the APPN entry point).

However, when deciding whether to route the request to SSCPs within NETA (the network identifier provided by HOSTC), HOSTD uses the DISJOINT network indicator provided by HOSTC to make the decision. If DISJOINT=YES was coded in HOSTC (and provided to HOSTD on the APPN search request), HOSTD routes the search request to other SSCPs within NETA, because it is assumed that the APPN network may provide the only connectivity between the two disjoint parts of network NETA. If DISJOINT=NO was coded (or allowed to default) in HOSTC, then HOSTD does not route the request to other SSCPs within NETA, because it is assumed that other connectivity exists between HOSTA and HOSTB (see Figure 131 on page 515). When HOSTD and HOSTC fail to locate the target LU, HOSTA continues to try other SSCPs and find the direct path to HOSTB.

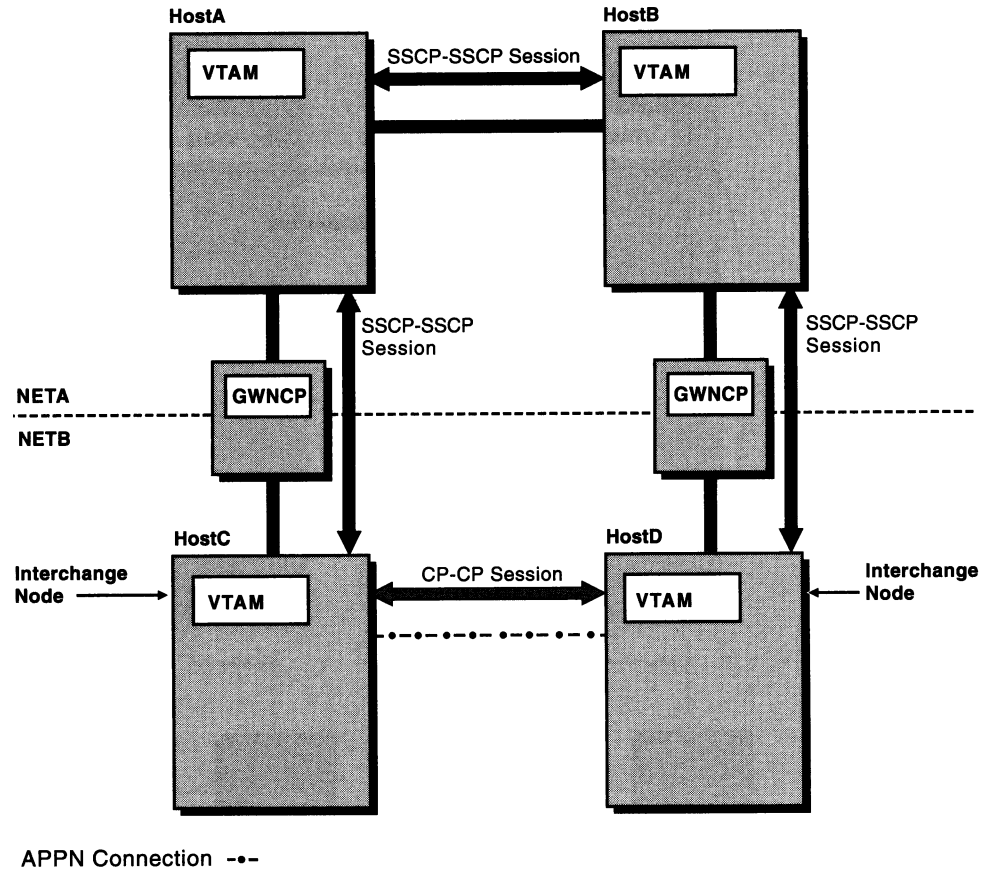


Figure 131. Hosts with Subarea Connection

Virtual-Route-Based Transmission Groups (VR-Based TGs)

A VR-based TG connects two APPN-capable VTAM nodes through a subarea network and allows them to act as APPN peers. The VR-based TG (whose TG number is always 255) then functions as any other TG in an APPN network. APPN CP-CP sessions can be established between these two nodes, which allows all APPN functions to be performed through the subarea network. For example, if the two nodes are attached to disjoint APPN subnetworks, the topologies of these two subnetworks are exchanged over the CP-CP sessions, thereby combining them into one large APPN network. Resource location and session establishment can also be performed through the subarea network using APPN protocols, rather than transforming APPN flows to subarea flows and back to APPN flows.

Although only one VR-based TG can be defined between any two VTAM nodes, this does not limit the use of this TG to only one virtual route (VR). In fact, the VR-based TG is used to represent all possible VRs between any two subareas in the domains of the two VTAMs. For example, in Figure 132 on page 516, if an application on HOSTA starts a session with an application on HOSTB, the VR-based TG may be selected as the route for this session. When this occurs, VR1 can be assigned as the underlying VR for the session, which will use the channel-to-channel connection between the two hosts. If an independent LU on network node A starts a session with an independent LU on network node D, the VR-based TG may be selected for a portion of the session route. However, in this case, VR2 can be assigned as the underlying VR, so that the direct connection between the two NCPs will be used.

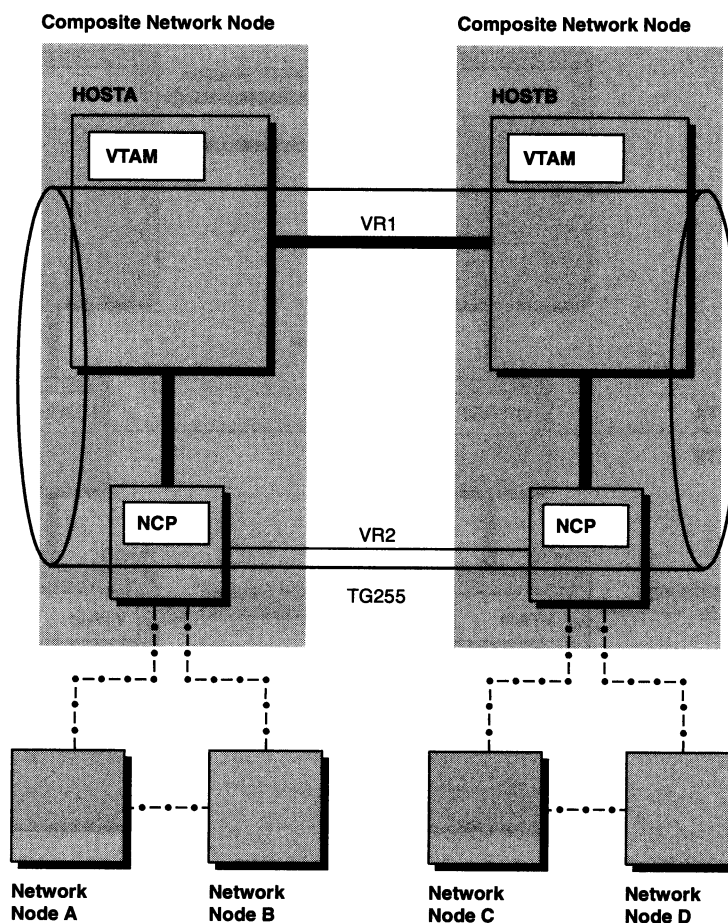


Figure 132. VR-Based TG Between Composite Network Nodes

Because an underlying VR is always assigned to sessions established using VR-based TGs, both APPN and subarea functions must be enabled on both nodes. This means that each node must be defined as an interchange node or a migration data host. (That is, the NODETYPE start option must be coded and each host must define a unique HOSTSA number.) Also, because a limited number of subarea flows must be sent between these two nodes to determine which underlying VR should be used, VR-based TG support also depends on a CDRM session being active between the two nodes.

Notes:

1. It is possible for a session route to be calculated which includes two or more consecutive VR-based TGs, as shown in Figure 133 on page 517. When this occurs, consecutive VR-based TGs are merged together into one VR-based TG. This is referred to as RSCV pruning. The resulting VR-based TG appears to directly connect the first and last VTAM nodes (the endpoints of the consecutive VR-based TGs), even though a VR-based TG was not actually defined between these two nodes. For this reason, it is not sufficient to have active CDRM sessions only between VR-based TG partner nodes. The existing subarea requirement that every SSCP in the network have an active CDRM session with every other SSCP in that network still holds.
2. VR-based TGs cannot be used across APPN or subarea network boundaries (that is, between nodes with different network IDs) or across APPN subnetwork boundaries created using the NATIVE=NO operand. For information on APPN

subnetwork boundaries created using the `NATIVE=NO` operand, see “APPN Multiple Network Connectivity” on page 545.

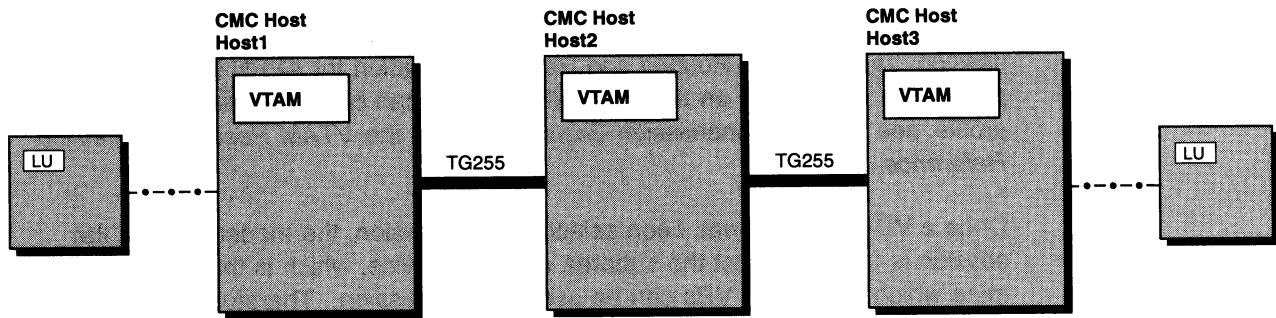


Figure 133. Multiple Contiguous VR-Based TGs

Defining a VR-based TG

To define a VR-based TG between two VTAM APPN nodes, code `VRTG=YES` as a start option at both VTAMs or on the CDRM definition statement for the adjacent VTAM. If `VRTG=YES` is coded at both VTAMs, then a VR-based TG is activated automatically when the CDRM session with the adjacent VTAM is established. If the VR-based TG connects two VTAM network nodes, a topology database update (TDU) is broadcast throughout the network, so that the VR-based TG is added to the topology database of all network nodes. If either VTAM is defined as an end node, then the VR-based TG is registered with the end node's network node server. Similarly, when the CDRM session between these two VTAMs terminates, the VR-based TG is deactivated and the appropriate nodes are notified that the TG is no longer active (either by sending another TDU through the network or by deleting a registered TG with an end node's network node server).

Note: To enable VR-based TGs with a specific adjacent CDRM without disrupting existing LU-LU sessions, use the `SAVESESS` operand when deactivating the adjacent CDRM. Keep in mind, however, that deactivation of a CDRM with `SAVESESS` disassociates the active sessions from the CDRM. Subsequent activations and deactivations of the CDRM have no effect on these sessions.

If there are no CP-CP sessions active between the two VTAM nodes, CP-CP session establishment is automatically initiated when a VR-based TG is activated. The CP-CP sessions may use the VR-based TG or any other APPN link between the two VTAMs that supports CP-CP sessions. If CP-CP sessions are not desired over a VR-based TG, code `VRTGCPCP=NO` as a start option or on the CDRM definition statement for the adjacent VTAM. Note, however, that an alternate CP-CP session path must exist between the two VTAMs, so that network traffic can be sent. For example, in Figure 132 on page 516, if a connection existed between network node B and network node C and CP-CP sessions were established between them, then `VRTGCPCP=NO` could be used to prevent CP-CP sessions from being established over the VR-based TG between the two VTAMs. When a session between two VTAM applications is initiated, the network traffic sent to locate the target resource would have to be sent through network node B and network node C, but the session could still be established over the direct VR-based TG connection.

The choice of which APPN TGs can be used for a given session is based on the APPN class of service and the characteristics defined for the links in the APPN

network. As such, APPN link characteristics must be defined for all VR-based TGs as well. This is done by coding the TGP operand on the adjacent CDRM definition statement, to specify the TG profile that describes the desired characteristics of the VR-based TG. For general information on how an APPN TG is chosen for a given session, see “Network Routing and Resource Location for APPN Nodes” on page 550. For details on the operands that you can code in a transmission group profile, see “APPN Transmission Group Profile” in the *VTAM Resource Definition Reference*.

Once a VR-based TG has been selected for a session, the modename for the session is used to select the subarea class of service, which is then used to determine which underlying VRs can be used for the session. Therefore, it is necessary to ensure that VRs and ERs are defined for every subarea class of service which may be selected when a VR-based TG is included in the route that is calculated during APPN session establishment.

VR-Based TG Recommendations

When defining the CP-CP sessions in an APPN network, it is important to remember that APPN networks do not require meshed CP-CP sessions the way subarea networks require meshed SSCP-SSCP sessions. In APPN networks, direct CP-CP sessions between all pairs of network nodes is not only impractical, due to the number of network nodes, it is not possible in many cases because CP-CP sessions are allowed with only physically adjacent nodes. Instead, there need only be a path of active CP-CP sessions between the origin and destination nodes in order to deliver a session establishment request, determine an appropriate session route, and establish the session.

In fact, too many CP-CP sessions in an APPN network may cause network nodes to spend too much time processing duplicate requests. For example, when network topology database updates and broadcast search requests are performed, every network node must send or receive every request over every pair of CP-CP sessions that it has with other network nodes. (This ensures that every network node receives the request.) For this reason, establishing CP-CP sessions with every adjacent node may also prove to be undesirable.

At the same time, not enough CP-CP sessions in an APPN network can cause portions of the network to become disjoint, when CP-CP sessions fail and network topology database updates and broadcast search requests can no longer be delivered to every network node. Therefore, some consideration must be given to the number of CP-CP sessions and the placement of those sessions, which takes into account a network's availability requirements. The ideal configuration is one that minimizes the number of CP-CP sessions while still ensuring that, at any point in time, every network node is logically connected to every other network node by a path of active CP-CP sessions.

The same considerations must be taken into account when defining VR-based TGs and CP-CP sessions through a subarea network. The following recommendations are provided to simplify the task of determining how many VR-based TGs and CP-CP sessions should be defined, and where in the network they should be located.

Physical Adjacency: VR-based TGs and CP-CP sessions should be defined between only physically adjacent VTAM nodes (or composite network nodes). They should not be defined through an intermediate VTAM. For example, in

Figure 133 on page 517, VR-based TGs and CP-CP sessions should only be defined between HOST1 and HOST2 and between HOST2 and HOST3. CP-CP sessions between HOST1 and HOST3 (through HOST2) are unnecessary, because session setup flows can be sent from HOST1 to HOST3 using HOST2 as an intermediate APPN node. Also, because RSCV pruning is performed, a two-hop VR-based TG route from HOST1 to HOST2 to HOST3 is pruned down to a one-hop VR-based TG route from HOST1 directly to HOST3, thereby allowing a VR-based TG to be used between HOST1 and HOST3, even though it was not explicitly defined.

CMC Connectivity: If the subarea network is set up as a communications management configuration (CMC) with multiple VTAMs acting as CMCs, then there must be enough VR-based TGs and CP-CP sessions defined between the CMCs to satisfy the network's availability requirements, based on the considerations described above. Although some redundant VR-based TGs and CP-CP sessions may have to be defined, to provide connectivity to all network nodes when TGs carrying CP-CP sessions fail abnormally, meshed VR-based TGs and CP-CP sessions between every pair of VTAM nodes will rarely be required, even when every VTAM is physically adjacent to every other VTAM.

Migration Data Host Connectivity: Migration data hosts are VTAM end nodes that also support CDRM sessions. As end nodes, migration data hosts require the services of a network node server in order to establish sessions with resources in other nodes. The network node server for VTAM end nodes is typically a VTAM network node or composite network node, possibly a CMC host.

If all of the migration data hosts are served by the same network node server, then VR-based TGs and CP-CP sessions need only be defined between each migration data host and its network node server (not between pairs of migration data hosts), as shown in Figure 134 on page 520. If a session is requested between two migration data hosts and the route calculated for the session is a two-hop VR-based TG route through the network node server, RSCV pruning will prune this route down to a one-hop VR-based TG route directly between the two migration data hosts and an appropriate VR will be selected for the session.

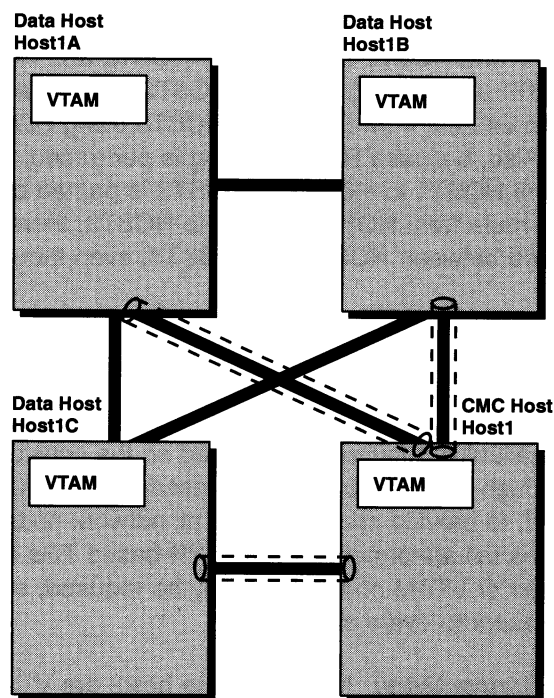


Figure 134. VR-Based TGs In a Communications Management Configuration

The same is true when the migration data hosts are served by different network node servers and there is a VR-based TG path between the two network node servers. In this case, each migration data host should define VR-based TGs to all potential network node servers. If a session is requested between two migration data hosts and the route calculated for the session is a multi-hop VR-based TG route through both network node servers, RSCV pruning will prune this route down to a one-hop VR-based TG route directly between the two migration data hosts and an appropriate VR will be selected for the session.

If two migration data hosts are served by different network node servers and there is not a VR-based TG path between the two network node servers, then each migration data host should define VR-based TGs to each adjacent VTAM network node and migration data host to improve the chances of finding a route between the migration data hosts that consists of entirely VR-based TGs.

APPN and Subarea Class of Service Resolution

For LU-LU session requests that use both APPN links and subarea virtual routes, it might be necessary for a given mode name to use different class-of-service names for the APPN links and the subarea virtual route. For this reason, logon mode table entries can specify an APPN class of service and a subarea class of service.

For each logon mode table associated with LUs that might be involved in a session with both APPN links and subarea virtual routes, specify an APPN class of service (by adding the APPNCOS operand) and a subarea class of service (using the COS operand) on the appropriate MODEENT macroinstructions. The subarea virtual routes use the class-of-service name coded on the COS operand, and APPN links use the class-of-service name coded on the APPNCOS operand. Using both the COS and APPNCOS operands enables you to use two different class of service names to represent the same class of service.

If neither the COS operand nor the APPNCOS operand is coded, the default class of service names (the unnamed class of service entry for subarea and the #CONNECT entry for APPN) are used.

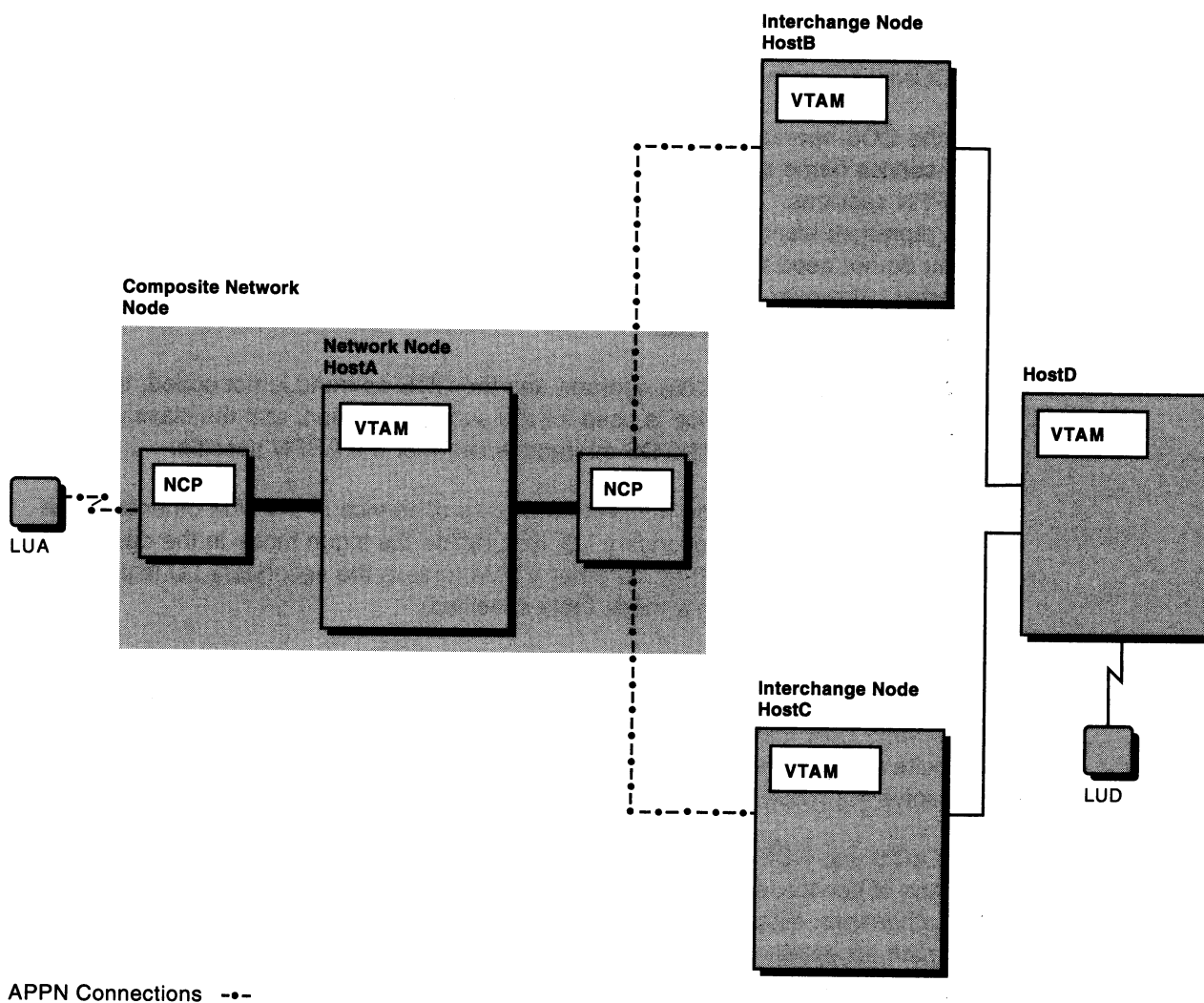
If the COS operand is coded and the APPNCOS operand is not coded, the class-of-service name specified on the COS operand is used for both the subarea and APPN sessions. Therefore, when you want to use the same class-of-service name to represent identical classes of service in both the APPN and subarea networks, you do not need to code the APPNCOS operand when the COS operand is already coded. However, make sure that a corresponding APPN class of service definition exists in all network nodes.

If you code the APPNCOS operand and the COS operand is not coded, the unnamed class of service is used for the subarea session, and the class of service designated on the APPNCOS operand is used for the APPN session.

If a logon mode must be resolved to a class of service in a VTAM other than the VTAM that owns the secondary LU, also define the logon mode in the default logon mode table (ISTINCLM) at the other VTAM (unless the secondary LU is predefined at the other VTAM with a mode table specified).

For example, in Figure 135 on page 522, LUA is in an APPN network and LUD is in a subarea network. An APPN class of service is used for the APPN link between LUA and the composite network node. Because routing within the composite network node is over a subarea virtual route, HOSTA must be able to resolve the mode name to a subarea class of service.

HOSTB and HOSTC must also be able to resolve the mode name to a subarea class of service, since either one of them can be used as the interchange node. Furthermore, since either HOSTB or HOSTC can be used as the interchange node, a path acceptable for the APPN class of service specified must be available between HOSTA and each interchange node.



APPN Connections ---

Figure 135. Class-of-Service Resolution at Multiple Interchange Nodes

Adding and Moving Nodes

To add or move workstations or applications in an APPN network, no coordinated LU definitions are required. As workstations and applications first become available or are later moved, APPN directory functions can locate them dynamically. If new hosts are added, only the resources on that host must be defined to that host.

If an APPN network node is to be moved from one APPN subnetwork to another, purge its topology database before activating CP-CP sessions in the new APPN subnetwork. This prevents broadcasting of network topology from the original subnetwork to the new subnetwork, information that is of no use in the new subnetwork. For a description of APPN subnetworks, see "APPN Multiple Network Connectivity" on page 545.

No definitions are necessary in the subarea network for resources added or moved in the APPN network. APPN directory functions can locate them dynamically.

Verifying a Combined APPN and Subarea Network

Verify a combined APPN and subarea network by following the steps in “Verifying a VTAM Network” on page 160, “Verifying a VTAM APPN Network” on page 560 , and “Verifying a Multiple-Domain Subarea Network” on page 435.

Chapter 10. Implementing an APPN Network

This chapter describes VTAM implementation issues specific to an APPN network. Use this chapter to implement VTAM in an APPN environment.

Before reading this chapter, you should be familiar with Chapter 5, "Getting Started with VTAM - Overview," and Chapter 6, "Getting Started with VTAM - Detail" (where appropriate).

For information on implementing APPN in a combined APPN and subarea network, see Chapter 9, "Implementing a Combined APPN and Subarea Network."

Implementing an APPN Network

An APPN network contains network nodes and end nodes connected by APPN connections. VTAM can be coded as either a network node or an end node. Configuring VTAM for APPN is optional. To configure VTAM as an APPN node, select the characteristics of the node and define the resources of that node to VTAM.

Starting VTAM with APPN Support

Support for APPN is implemented by defining the NODETYPE start option. If you do not define the NODETYPE start option to specify APPN function, VTAM operates only as a subarea node. Start options for APPN support do not default unless you have requested APPN support.

The value defined for the NODETYPE start option indicates whether VTAM functions as an end node or a network node. There is no default for the NODETYPE start option. If the NODETYPE start option is not specified, VTAM operates only as a subarea node.

Enabling Control Sessions

The CPCP start option specifies whether an APPN node supports CP-CP sessions with an adjacent node. The CPCP start option is valid only for nodes at which the NODETYPE start option is specified. APPN nodes use CP-CP sessions to communicate topology, routing, directory, and session information to adjacent control points. If you use the default, CP-CP sessions are supported on all APPN leased links. Because CP-CP sessions carry network control information, they should be active as long as the node is active. Using the default allows you to control whether your APPN switched links support CP-CP sessions on a link-by-link basis by defining the CPCP operand.

You can use the CPCP start option to specify support for CP-CP sessions on leased links only, on switched links only, on both leased and switched, or on neither leased nor switched.

The CPCP operand can be coded on the PU definition statement for the local SNA, model, NCP, packet, channel-attachment, LAN, and switched major nodes.

CP-CP Sessions Between Two VTAM Nodes

CP-CP sessions between two VTAM nodes requires both CDRSC and ADJCP definitions at each VTAM to represent the partner CP. The CDRSC definition represents the partner's LU and the ADJCP definition represents the adjacent CP at the connection level. ADJCP definitions can be created dynamically or predefined. When needed, CDRSC definitions are dynamically created for adjacent CPs, regardless of the value coded for DYNLU.

Defining Adjacent APPN Nodes

The DYNADJCP start option controls whether adjacent CP minor nodes can be created dynamically. An adjacent CP major node, ISTADJCP, is automatically created when a VTAM node is activated. An adjacent CP minor node describes the CP name, the network identifier, node type (end node or network node), and whether dynamic definition is allowed for LUs owned by this adjacent node.

When the NODETYPE start option is specified, the default for the DYNADJCP start option is DYNADJCP=YES. If you allow the DYNADJCP start option to default to YES, adjacent CP minor nodes are created as needed to provide control and management of connections to adjacent APPN nodes.

Note: Unless CDRSCs are predefined for adjacent CPs, CDRDYN=YES is also required for the dynamic creation of adjacent CP minor nodes.

If you define the DYNADJCP start option as NO, you need to define every potential adjacent CP within adjacent CP major and minor nodes. Connections are established with only those nodes you specify. If you do not allow dynamic ADJCPs, and you do not define any adjacent CPs with the ADJCP definition statement, all connection attempts will fail.

You can choose to allow dynamic ADJCP minor nodes to be created for some links and not allow them for others. Be aware that for each link where you restrict dynamic definition, you need to define each CP to which you will allow connections with an ADJCP definition statement.

If you want dynamic definitions on most links, you can define the DYNADJCP start option as YES, and override it by defining the DYNADJCP operand as NO for the links on which you want to restrict dynamic definitions.

If you want limited dynamic definitions, you can code the DYNADJCP start option as NO and override it by coding the DYNADJCP operand as YES for links on which you want dynamic definitions.

Defining the Logon Mode for CP-CP Sessions

CP-CP sessions always use the CPSVCMG logon mode. This logon mode is defined in the IBM-supplied default logon mode table, ISTINCLM. If you use your own logon mode table instead of the IBM-supplied table, be sure you include a CPSVCMG logon mode entry in it. See ISTINCLM for the recommended values for the CPSVCMG logon mode entry.

If no CPSVCMG logon mode is defined, CP-CP session activation will fail, and VTAM will issue the following message:

```
IST264I REQUIRED LOGMODE NAME CPSVCMG UNDEFINED.
```

In addition, message IST663I will display the error sense code X'08210002' (invalid mode name at CP).

Connecting an APPN Node to the Network

For a node to communicate with other nodes as a peer, rather than as a participant in a hierarchical relationship, the node must be aware of other APPN nodes. VTAM nodes use a combination of the following available definitions and information to learn about adjacent nodes:

- Definition of the physical line or port
- Definition of the adjacent link station, either defined dynamically or predefined with PU definition statements
- Transmission group number
- Adjacent control point name.

In addition to this information, before and during link activation between adjacent link stations, nodes obtain information about the capability of an adjacent node through a message unit called an exchange identification (XID), which is used to convey node and link characteristics. XIDs are exchanged by adjacent nodes to establish and negotiate link and node characteristics. After link activation, XIDs are used to communicate changes in these characteristics and to negotiate the TG number used to represent the connection.

APPN nodes are connected using type 2.1 connections. APPN connections provide the same benefits as LEN connections. With APPN connections, you do not need to define CDRMs, PATH definitions, or ADJSSCP tables. You can dynamically define switched connections and independent logical units and the connectivity of applications and dependent LUs is not limited.

Connections through Boundary Function-Based Transmission Groups

Nodes are connected by transmission groups (TGs). A transmission group connects a pair of adjacent link stations across the transmission medium. A boundary function (BF)-based APPN TG is used to connect two APPN nodes that use the NCP or VTAM boundary function.

Transmission groups are described with standard APPN characteristics:

- Cost-per-connect time
- Cost-per-byte
- Security
- Propagation delay
- Effective capacity
- User-defined values.

The values specified for these characteristics are saved in the topology database and are used to determine the weight of the TG for route calculation. The values you specify represent the characteristics of the link; the values do not affect the actual physical characteristics of the link. If the physical characteristics of the link change, the values specified for the TG characteristics should probably be changed to reflect the new physical characteristics. The topology database is discussed further under "Topology Database" on page 16.

By default, all TGs assume the same characteristics. To distinguish one TG from another, TG characteristics can be specified:

- As transmission group profiles, which are groups of TG characteristics that can be applied to several PUs
- On definition statements.

The characteristics of TGs owned by other nodes are learned through topology database updates (TDUs).

Multiple Connections with Parallel Transmission Groups

Parallel TGs for multiple connections between APPN nodes are supported. This support means that a node can have multiple links to the same adjacent node. Parallel TGs have different TG numbers and can have different TG characteristics assigned to them. Multiple-link TGs (many links with one TG number) are not supported in APPN by VTAM. The support for parallel TGs matches that of most other APPN products.

If you have parallel TGs with CONNTYPE=APPN coded on the adjacent link stations, you can establish one or more APPN connections or one LEN connection between the two nodes. You cannot combine LEN and APPN connections or establish more than one LEN connection to the same adjacent node over APPN-capable TGs.

Support for multiple connections allows an independent LU in the APPN network to establish sessions through or into the subarea network by using several boundary-function TGs. An LU can have multiple sessions with one or more LUs through multiple boundary-function TGs and through different physical paths that function as parallel TGs. This support is also available for TGs attached to LEN nodes.

When coding CONNTYPE=APPN (or letting it default to that value), XID exchange rules for parallel TGs are enforced. This means that if you have PUs with duplicate CPNAMES, you should resolve this duplication before that node becomes an APPN node. If the node is to remain a LEN node then CONNTYPE=LEN should be coded on the PU statement. Coding CONNTYPE=LEN results in VTAM avoiding the test for duplicate CPNAMES.

Connection Network

A connection network is a representation of a shared access transport facility (SATF), such as a local area network (LAN), that enables nodes identifying their connectivity to the SATF by a common virtual routing node to communicate without having individually defined connections to one another. This is illustrated in the following figures.

In Figure 136 on page 529, when a LU-LU session between resources at AS400A and HOSTB is set up, the optimal route is directly through the token ring between the two nodes. However, if AS400A and HOSTB are not directly defined to each other or are not connected to the same connection network, the indirect route is selected through HOSTA.

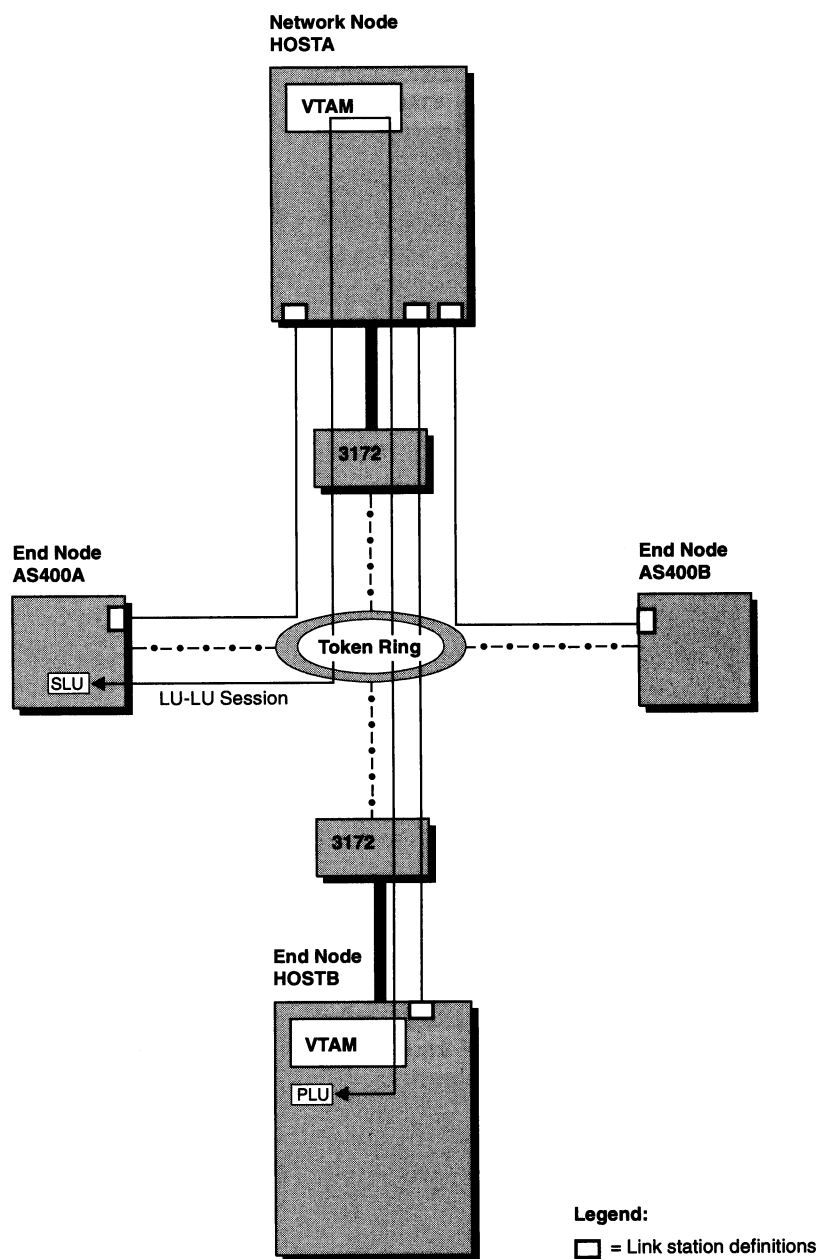


Figure 136. VTAM Attachment to a LAN - No Meshed Connection Definitions

Optimal route calculation is achieved in one of two ways:

- Meshed connection definitions
- Connection network definition

A definition for the connection between each pair of nodes ($n*(n-1)$ definitions, where n equals the number of nodes) enables optimal route calculation between all nodes on the SATF. In Figure 137 on page 530, for example, the pictured LU-LU session no longer traverses HOSTA.

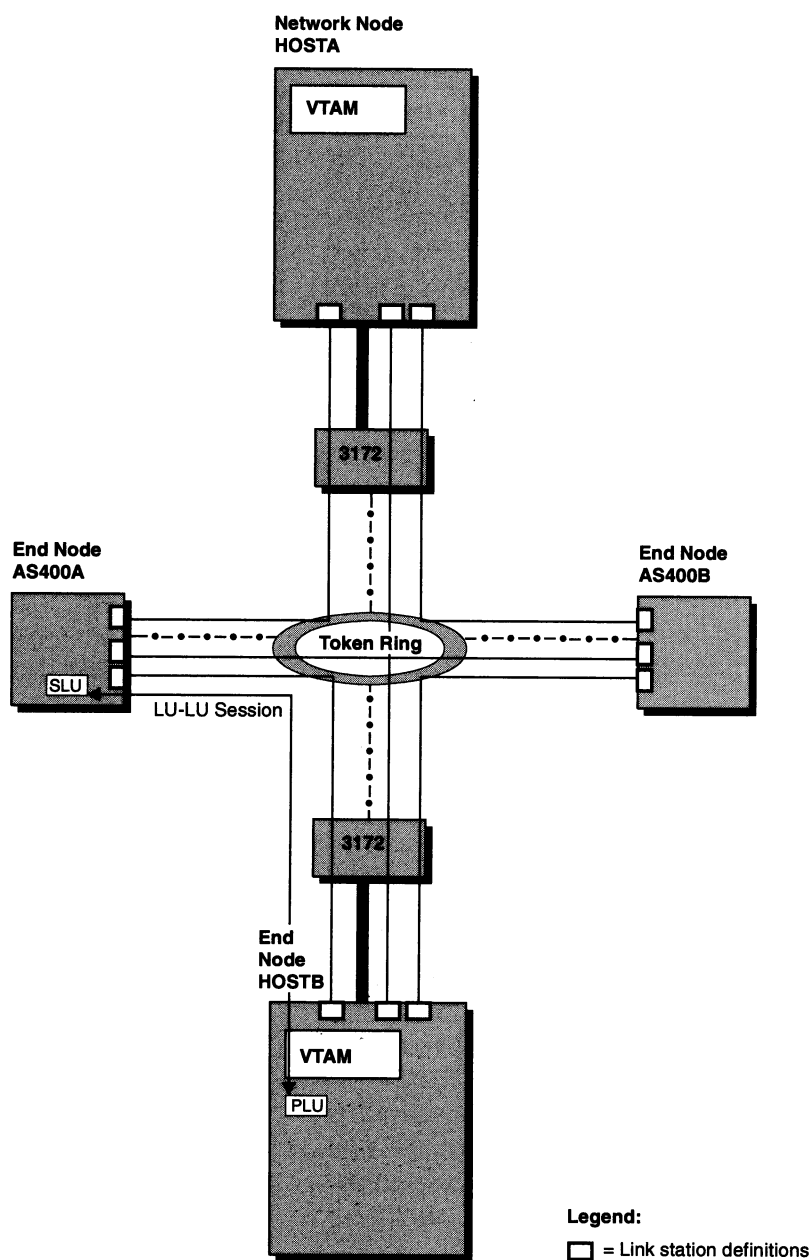


Figure 137. VTAM Attachment to a LAN - Meshed Connection Definitions Provide Optimal Route Calculation

In a large network, however, this system definition is extensive. System definition is greatly reduced by defining a connection network to represent the shared access transport facility (here, the token ring). In a connection network, end nodes need only define a connection to a virtual node, as shown in Figure 138 on page 531.

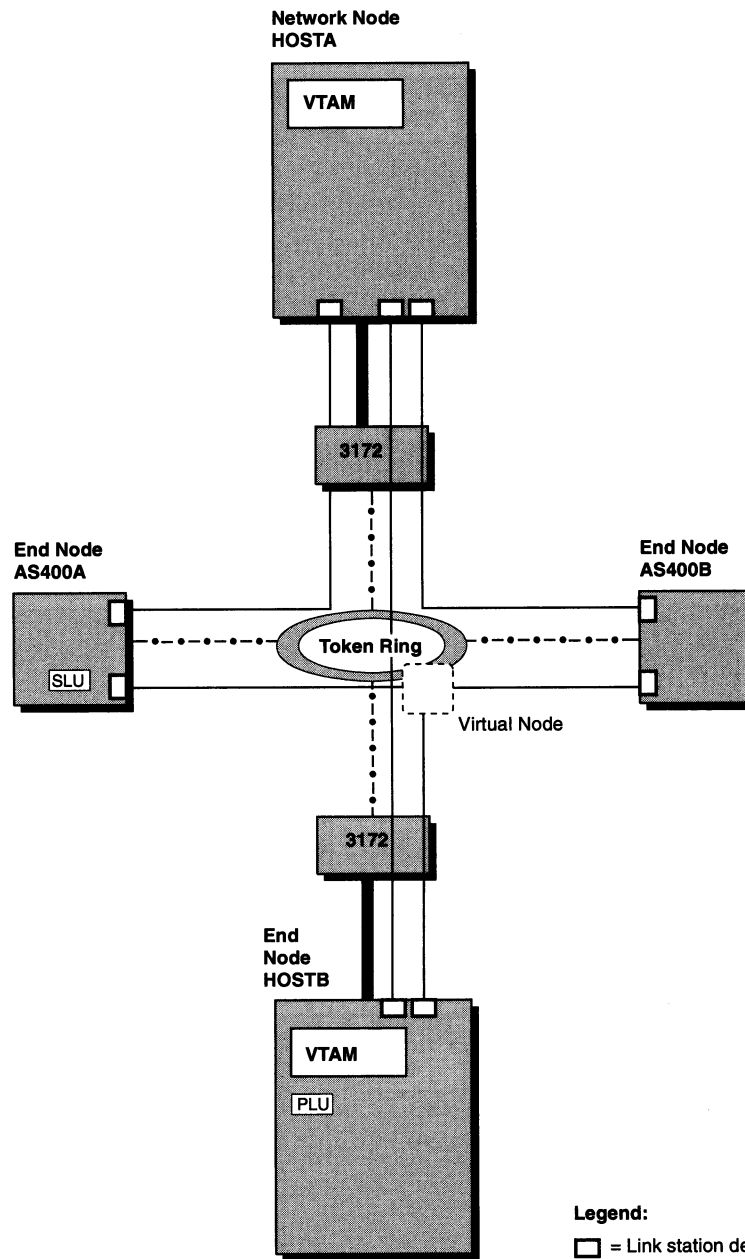


Figure 138. VTAM Attachment to a Connection Network Reduces Required Connection Definitions

The virtual node is reported to the topology database and can be chosen as the intermediate node during route calculation. As shown in Figure 139 on page 532, BINDs can then be routed directly to the destination node over a dynamically created TG. Thus, the definition of a connection network to represent the token ring reduces required system definition and enables optimal route calculation.

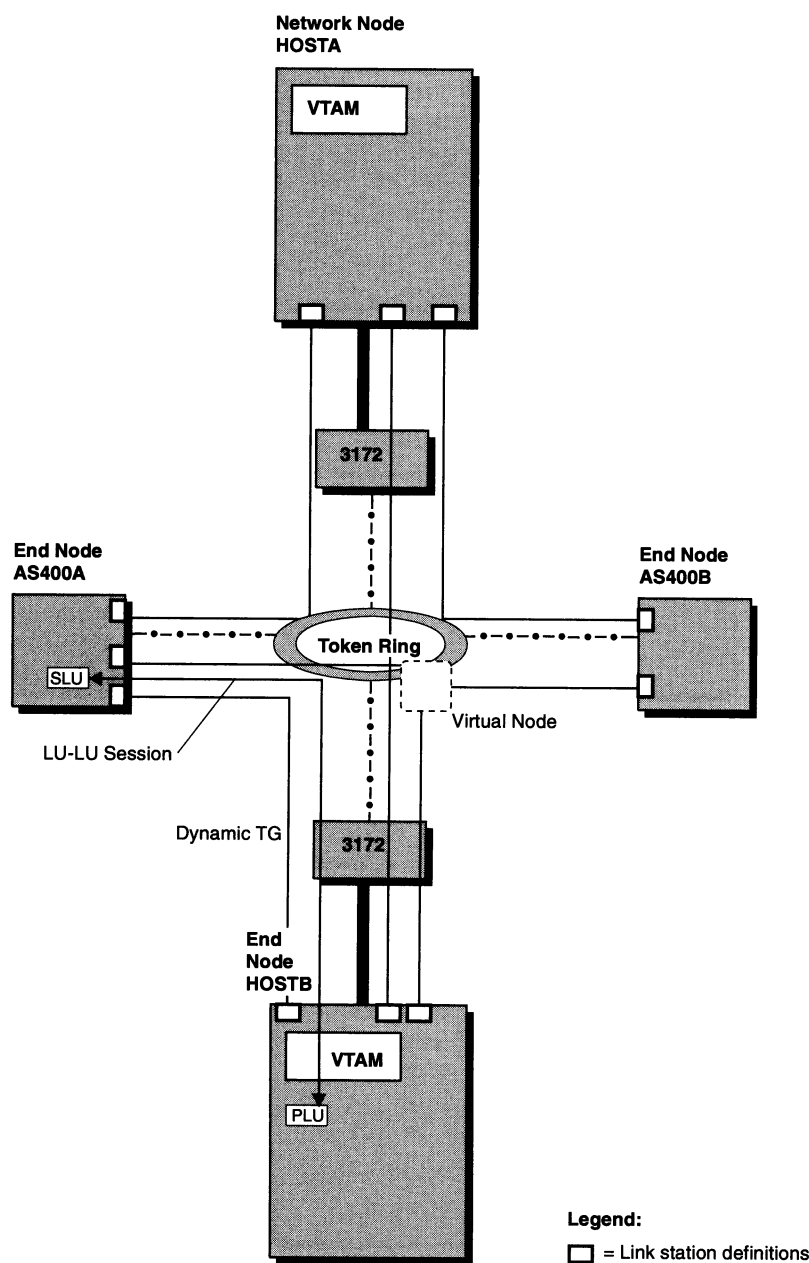


Figure 139. VTAM Attachment to a Connection Network Also Enables Optimal Route Calculation

To define a connection to a connection network, include the following in the NCP major node, the XCA major node, or the LAN major node.

- In the NCP major node (NTRI physical lines) code the:
 - VNNNAME and VNGROUP operands on the LINE definition statement. The VNNNAME operand specifies the CPNAME of the virtual node. The VNGROUP operand specifies the name of the logical group containing dial-out links through the connection network named on the VNNNAME operand.

Note: DYNPU=YES is the default when VNNAME and VNGROUP are coded. For more information on DYNPU, see “Defining Switched Resources Dynamically” on page 143.

- Optional operands on the LINE definition statement include the TGP operand to indicate a TG profile containing TG characteristics, or operands for individual TG characteristics.
- In the XCA major node (3172 interconnect controller lines), code the:
 - VNNAME and VNGROUP operands on the PORT definition statement. The VNGROUP operand specifies the GROUP in this major node containing the lines for the virtual node named on the VNNAME operand.
 - Note:** DYNPU=YES is the default when VNNAME and VNGROUP are coded. DIAL=YES is required on the GROUP named on the VNGROUP operand.
 - Optional operands on the PORT definition statement include the TGP operand to indicate a TG profile containing TG characteristics, and operands for individual TG characteristics,
- **VM,VSE** In the LAN major node, code the:
 - VNNAME and VNGROUP operands on the PORT definition statement. The VNGROUP operand specifies the GROUP in this major node containing the lines for the virtual node named on the VNNAME operand.
 - Note:** DYNPU=YES is the default when VNNAME and VNGROUP are coded. DIAL=YES is required on the GROUP named on the VNGROUP operand.
 - Optional operands on the PORT definition statement include the TGP operand to indicate a TG profile containing TG characteristics, and operands for individual TG characteristics,

Notes:

1. NCP Version 7 Release 1 is required for NCP/Token-Ring interconnection (NTRI) support of connection network.
2. If CP-CP sessions are required between two nodes on the shared access transport facility, the dialing-out node must define the PU for any node it is to call; routes traversing a virtual node cannot be used for CP-CP sessions.
3. An ADJCP definition statement with the VN=YES operand can optionally be placed into the adjacent control point major node. The VN=YES operand indicates that this adjacent CP is a virtual node. The VN operand cannot be specified when the NN operand or the DYNLU=YES operand is also specified.
4. A connection network cannot be used for connectivity across a network or sub-network boundary.

Defining an APPN Connection

By coding the NODETYPE start option, you specify a node as an APPN node. After NODETYPE is coded, the following start options default to the values needed to enable APPN connections.

CPCP

The CPCP start option defaults to LEASED and enables CP-CP control sessions across APPN leased links. The other possible values for CPCP are SWITCHED, YES (for all

links), and NO (for no links). For information on modifying CP-CP session enablement, see “Enabling Control Sessions” on page 525.

CONNTYPE

The CONNTYPE start option defaults to the value APPN and enables full APPN function across a link. The other connection type value is LEN. There is also a CONNTYPE operand on the PU definition statement that can be used to override the value of the start option for specific links. For more information on the CONNTYPE start option and operand, see “Implementing a Combined APPN and Subarea Network” on page 497.

LU definitions are not needed if you use dynamic independent LUs. See “Defining Independent LUs” on page 220.

Coding examples for each of the connection types are included in this section following the general requirements. See “Using Type 2.1 Channel Connections between APPN Nodes” on page 535, “Using Leased Connections between APPN Nodes” on page 537, and “Using XCA over a 3172 Interconnect Controller between APPN Nodes” on page 539.

If you take advantage of the defaults, you only need to consider the following requirements for an APPN connection:

- To enable CP-CP sessions over a switched APPN connection, the CPCP start option should be defined as YES or SWITCHED.
- The PU definition for the NCP in a composite network node or for the VTAM in a network node or end node needs to indicate exchange identification support (XID=YES). (For a switched connection, XID cannot be coded but is automatically set to YES.)
- Use different names for the CP name and the PU name of a node.
- If you choose to allow dynamic definition of your independent logical units by defining the DYNLU start option or operand as YES, then you only need to define the independent LUs on their owning CP.
- If you choose not to allow dynamic definition of your independent logical units, each network node that needs to connect to the independent logical unit needs a CDRSC definition for the resource. If you chose to code the ALSLIST operand on the CDRSC statement, the ALSLIST operand should specify either the name of an adjacent link station that can be used to connect to the LU or ISTAPNPU. ISTAPNPU is a generic representation for an APPN adjacent link station. If ISTAPNPU is included in the ALSLIST, VTAM uses any available APPN link station when calculating a route to the LU.

If you chose not to code ALSLIST, VTAM searches for the LU in the APPN network and, if found, adds ISTAPNPU to the ALSLIST dynamically.

Also, unlike a LEN adjacent link station, an APPN adjacent link station is not required to have a NETID that matches the NETID of the independent LU at session establishment.

- To establish a statically defined switched APPN connection either the CPNAME operand or the IDBLK and IDNUM operands must be defined for each switched PU definition. If the CPNAME operand is defined, its value must be the same as the CP name of the node (the SSCPNAME start option for VTAM nodes)

that owns the PU. If the values are not the same, VTAM cannot locate the PU and LU definitions. There may be more than one PU related to that same CP.

To establish a statically defined switched LEN connection either the CPNAME operand or the IDBLK and IDNUM operands must be defined for each switched PU definition. If the CPNAME operand is defined, its value specifically identifies a specific PU and can not be duplicated.

- To establish a dynamically defined switched connection, you can allow the DYNADJCP start option to default to YES and code the DYNPU operand on the GROUP definition statement in the XCA or NCP major node.
- You need to know the maximum number of sessions that will be established through each NCP in a composite network node. For NCP type 2.1 connections, this number is defined during NCP generation and cannot dynamically expand. However, for a direct VTAM connection, the session control blocks are held in the BSBUFF buffer pool that can expand and contract as necessary.
- For an NCP connection, you should also know the maximum number of sessions that any particular type 2.1 node can establish per link station. If you do not specify this limit, a single independent LU in a type 2.1 node can exhaust all of the session control blocks generated in the NCP. For information on specifying this limit, see “Limiting Sessions” on page 227.

There is no corresponding control for type 2.1 nodes directly attached to a VTAM. However, because this is a buffer pool, it can expand as needed to accommodate more sessions.

- When an independent LU in a type 2.1 node acts as the primary logical unit, NCP must assign another network address to the PLU. This address pool is generated in NCP. Because this is a predefined pool, you should know the maximum number of PLU sessions that all type 2.1 node independent LUs attached to that NCP can establish concurrently. For a direct VTAM connection, the addresses are assigned from the element pool that is managed by VTAM.

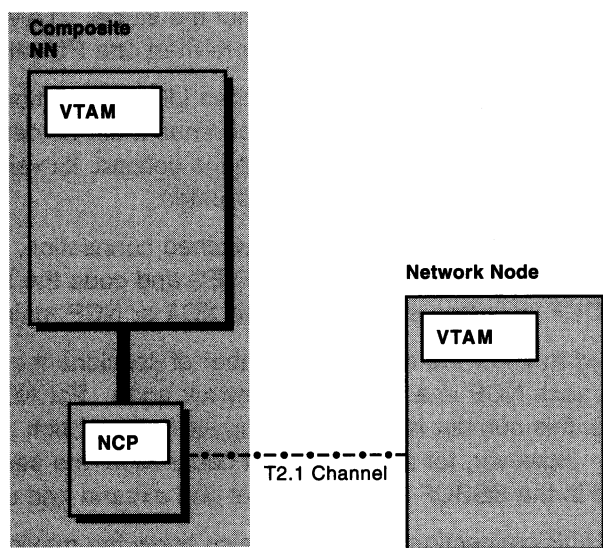
The following subsections contain examples of the definition statements needed for an APPN connection:

- “Using Type 2.1 Channel Connections between APPN Nodes”
- “Using Leased Connections between APPN Nodes” on page 537
- “Using XCA over a 3172 Interconnect Controller between APPN Nodes” on page 539.

Using Type 2.1 Channel Connections between APPN Nodes

Type 2.1 channel connections can be used to connect two network nodes, or a network node and an end node. The following example shows how to connect an NCP in a composite network node to a network node.

Note: This sample does not include all definitions needed for an NCP major node. For further information on defining the NCP to VTAM, see “Attaching NCP to VTAM” on page 138.



APPN Connection -.-

Figure 140. Type 2.1 Channel Connection between a Composite Network Node and a Network Node

Composite Network Node Definitions: Defining the NCP Major Node

```

:
GRP3AAA1 GROUP LNCTL=CA,CA=TYPE5,NCPCA=ACTIVE
LN3AAA1 LINE ADDRESS=04,TIMEOUT=840.0,CASDL=420.0,TRANSFR=254,      X
                INBFRS=128,ANS=CONT
P3A21AA1 PU    PUTYPE=2,XID=YES,                                     X
                CONNTYPE=APPN,CPCP=YES
:

```

Network Node Definitions: Defining the Local SNA Major Node

```

LSNA3AA VBUILD TYPE=LOCAL
LSNA3APA PU    PUTYPE=2,CUADDR=050,XID=YES,                          *
                SSCPFM=USSSCS,MAXBFRU=15,                            *
                CONNTYPE=APPN,CPCP=YES

```

APPN host-to-host channel connections enable two VTAMs to communicate using APPN protocols over MPC connections. Each VTAM must:

- Define the channel connection to the adjacent VTAM as an APPN PU using the PU definition statement with the TRLE operand in a local SNA major node. The TRLE operand specifies the transport resource list element (TRLE) containing transport characteristics of the PU.
- Define the transport resource list element using the TRLE definition statement in the transport resource list (TRL) major node (ISTTRL). A TRLE defines a multipath channel connection that can be used as an APPN connection.

After creating the local SNA major node and TRL major node in each VTAM, you can use an APPN host-to-host channel connection by following the steps below for each VTAM:

1. Activate the TRL major node.
2. Activate the local SNA major node.

3. Activate the PU that defines the channel connection to the adjacent VTAM as an APPN PU.

One TRL major node exists in VTAM. The TRL major node can be dynamically modified using V NET,ACT,ID= with the UPDATE operand.

If UPDATE=ALL is specified, the VTAMLST member designated with the ID operand replaces the existing TRL major node. However, note that TRLEs in use are not deleted and remain in the TRL major node along with the new TRLEs.

If UPDATE=ADD is specified, the TRLE entries in the VTAMLST member identified on the ID operand are added to the TRL major node.

Figure 141 shows two network nodes connected via an APPN host-to-host channel. Following Figure 141 are corresponding sample definitions.

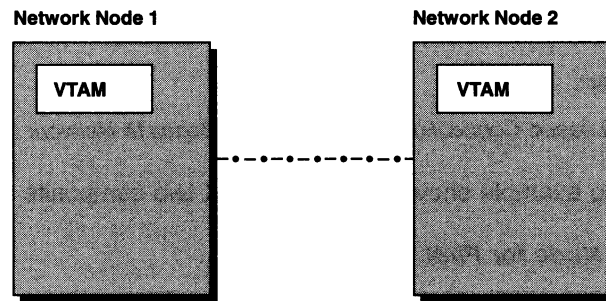


Figure 141. APPN Host-to-Host Channel Connection

TRL Major Node for Network Node 1

```

TRLNN1  VBUILD TYPE=TRL
MPCLN1  TRLE  LNCTL=MPC,MAXBFRU=16,          *
          READ=(BCA),WRITE=(BCB)
  
```

Local SNA Major Node for Network Node 1

```

TRLLCL1 VBUILD TYPE=LOCAL
TRLPU1  PU    TRLE=MPCLN1,ISTATUS=ACTIVE,VPACING=0,  *
          SSCPFM=USSSCS,CONNTYPE=APPN,CPCP=YES
  
```

TRL Major Node for Network Node 2

```

TRLNN2  VBUILD TYPE=TRL
MPCLN2  TRLE  LNCTL=MPC,MAXBFRU=16,          *
          READ=(BCB),WRITE=(BCA)
  
```

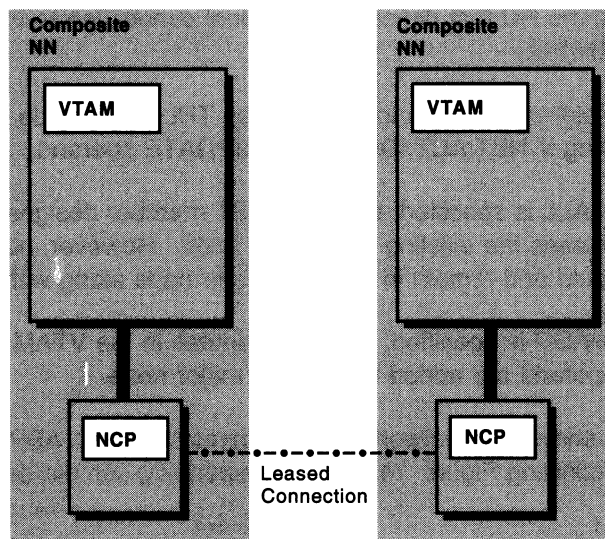
Local SNA Major Node for Network Node 2

```

TRLLCL2 VBUILD TYPE=LOCAL
TRLPU2  PU    TRLE=MPCLN2,ISTATUS=ACTIVE,VPACING=0,  *
          SSCPFM=USSSCS,CONNTYPE=APPN,CPCP=YES
  
```

Using Leased Connections between APPN Nodes

A leased connection can be used to connect a composite network node to a composite network node or to other nodes that support leased connections.



APPN Connection -.-

Figure 142. Leased Connection between Two Composite Network Nodes

The following example shows how to connect two composite network nodes.

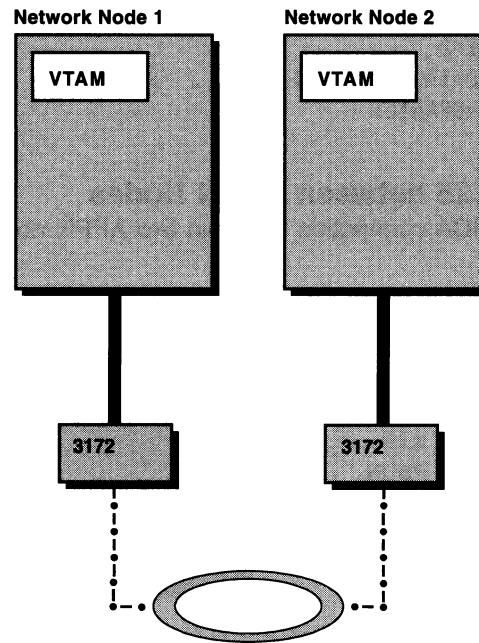
NCP Major Node for First Host

```
GRP4A6  GROUP LNCTL=SDLC,DIAL=NO,MODE=SEC,TYPE=NCP,          X
          CLOCKNG=EXT,DUPLEX=HALF,SPEED=1200
*
LN4A6   LINE  ADDRESS=04E, TADDR=C1
*
          SERVICE ORDER=(P4A4956C)
*
P4A4956C PU  PUTYPE=2,XID=YES,SSCPFM=USSSCS,                X
          MAXDATA=256,MAXOUT=1,PASSLIM=1,                    X
          ANS=CONTINUE,CONNTYPE=APPN,CPCP=YES
```

NCP Major Node for Second Host

```
GRP3A9  GROUP LNCTL=SDLC,DIAL=NO,TYPE=NCP,                  X
          CLOCKNG=EXT,DUPLEX=HALF,SPEED=1200
*
LN3A11  LINE  ADDRESS=03F
*
          SERVICE ORDER=(P3A4956M)
*
P3A4956M PU  PUTYPE=2,ADDR=C1,XID=YES,SSCPFM=USSSCS,        X
          MAXDATA=256,MAXOUT=1,PASSLIM=1,                    X
          ANS=CONTINUE,CONNTYPE=APPN,CPCP=YES
```

Using XCA over a 3172 Interconnect Controller between APPN Nodes



APPN Connection --

Figure 143. Two Network Nodes Connected Using XCA over a 3172 Interconnect Controller

A 3172 connection can be used to connect a composite network node, end node, or network node with any other type of APPN node. The following example shows how to connect two network nodes.

By coding DYNPU in the XCA major node, network node 1 does not need a switched major node; however, network node 2 must initiate the session using the DIALNO coded in its switched major node.

XCA Major Node in Network Node 1

```
XCA1A1L  VBUILD  TYPE=XCA
PORT1A1L  PORT    MEDIUM=RING,ADAPNO=1,SAPADDR=4,CUADDR=500,TIMER=254
GP1A1L1   GROUP   DIAL=YES,ANSWER=ON,CALL=INOUT,                      X
                                DYNPU=YES,DYNPUPFX=BA
L1A1L1AA  LINE
P1A1L1AA  PU
```

XCA Major Node in Network Node 2

```
XCAA1L  VBUILD  TYPE=XCA
PORTAA1L  PORT    MEDIUM=RING,ADAPNO=1,SAPADDR=4,CUADDR=503,TIMER=254
GPAA1L1   GROUP   DIAL=YES,ANSWER=ON,CALL=INOUT
LAA1L11A  LINE
PAA1L11A  PU
```

Switched Major Node in Network Node 2

```

SWXCAA VBUILD TYPE=SWNET,MAXNO=256,MAXGRP=256
SWP1A1 PU    MAXPATH=5,MAXDATA=256,ADDR=11,CPNAME=SSCP1A,      X
          PUTYPE=2
PTH1A1L PATH DIALNO=1A04003A11111111,                          X
          GRPNM=GPA11L1
  
```

Using ICA over X.25 between APPN Nodes

Figure 144 shows an ICA connection between two APPN nodes over a X.25 network.

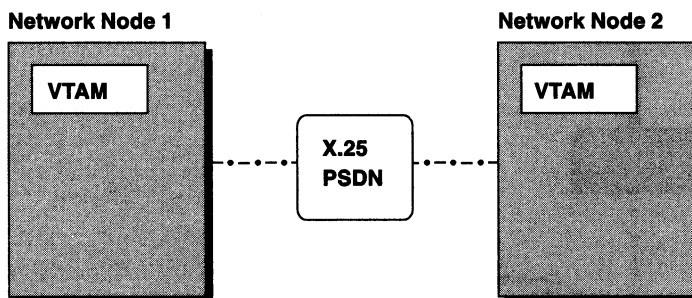


Figure 144. Using ICA over X.25 between APPN Nodes

Packet Major Node in Network Node 1

```

PKT1AP1 VBUILD TYPE=PACKET
PORT1AP1 PORT CUADDR=(083,08B),NETTYPE=8,NETLEVEL=84,PWINDOW=7,  *
          VCALLS=(,004,007,,),DIALNO=8470918,MAXOUT=7,PMOD=8,    *
          REPLYTO=25.5,RETRIES=31
*
P1PARMS VCPARMS LC=(001,003),PLENGTH=128,PWINDOW=7
*
* NON-SWITCHED GROUP (PVC)
*
P11AGP01 GROUP DIAL=NO,XID=YES,LNCTL=SDLC
P11ALN01 LINE  ADDRESS=001,XID=YES,ISTATUS=INACTIVE
P11APU01 PU    ADDR=01,PUTYPE=2,XID=YES,ISTATUS=INACTIVE
P11ALU01 LU    LOCADDR=01
*
* SWITCHED GROUP (SVC)
*
P11AGP02 GROUP DIAL=YES,LNCTL=SDLC,SUBADIAL=NO,ISTATUS=INACTIVE
P11ALN02 LINE  ISTATUS=INACTIVE,CALL=INOUT
P11APU02 PU    ISTATUS=INACTIVE
  
```


Packet Major Node in Network Node 2

```

PKT2AP1  VBUILD TYPE=PACKET
PORT2AP1  PORT  CUADDR=(087,08F),NETTYPE=8,NETLEVEL=84,PWINDOW=7,      *
              VCALLS=(,004,012,,),DIALNO=8488214,MAXOUT=7,PMOD=8,      *
              REPLYTO=25.5,RETRIES=31
*
P1PARMS  VCPARMS LC=(001,003),PLENGTH=128,PWINDOW=7
*
* NON-SWITCHED GROUP (PVC)
*
P12AGP01 GROUP DIAL=NO,XID=YES,LNCTL=SDLC
P12ALN01 LINE ADDRESS=001,XID=YES,ISTATUS=INACTIVE
P12APU01 PU   ADDR=01,PUTYPE=2,XID=YES,ISTATUS=INACTIVE
P12ALU01 LU   LOCADDR=01
*
* SWITCHED GROUP (SVC)
*
P12AGP02 GROUP DIAL=YES,LNCTL=SDLC,SUBADIAL=NO,ISTATUS=INACTIVE
P12ALN02 LINE ISTATUS=INACTIVE,CALL=INOUT
P12APU02 PU   ISTATUS=INACTIVE

```

Switched Major Node in Network Node 1

```

SWNDX1A1 VBUILD TYPE=SWNET,MAXNO=8,MAXGRP=8
*
*****
* DIAL IN/OUT GROUPS
*****
SWPX1AP1 PU   ADDR=B1,MAXPATH=5,MAXDATA=256,                          X
              PUTYPE=2,CAPACITY=10K,CPNAME=SSCP2A
PTHX1AP1 PATH DIALNO=8488214,                                          X
              GID=1,PID=1,REDIAL=0,                                    X
              GRPNM=P11AGP02

```

Switched Major Node in Network Node 2

```

SWNDX2A1 VBUILD TYPE=SWNET,MAXNO=8,MAXGRP=8
*
*****
* DIAL IN/OUT GROUPS
*****
SWPX2AP1 PU   ADDR=C1,IDNUM=00001,MAXPATH=5,MAXDATA=256,              X
              PUTYPE=2,CAPACITY=10K,CPNAME=SSCP1A
PTHX2AP1 PATH DIALNO=8470918,                                          X
              GID=1,PID=1,REDIAL=0,                                    X
              GRPNM=P12AGP02

```

Using ICA over a Token Ring between APPN Nodes

Figure 145 shows an ICA connection between two APPN nodes over a token-ring network.

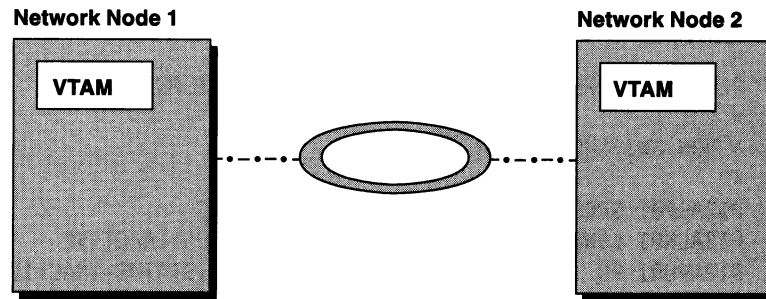


Figure 145. Using ICA over Token Ring between APPN Nodes

LAN Major Node in Network Node 1

```
A01LAN  VBUILD TYPE=LAN
PORTA01  PORT  CUADDR=B00, LANCON=(25.5,5), MACADDR=400000092211,      C
          MAXDATA=2012, MAXSTN=9, SAPADDR=4

*
GP1A2A  GROUP LNCTL=SDLC, DIAL=YES, ANSWER=ON, CALL=INOUT
LANLINE1 LINE  ISTATUS=INACTIVE
LANPU1  PU
```

LAN Major Node in Network Node 2

```
A02LAN  VBUILD TYPE=LAN
PORTA02  PORT  CUADDR=B20, LANCON=(25.5,5), MACADDR=400000092212,      C
          MAXDATA=2012, MAXSTN=9, SAPADDR=4

*
GP2A1A  GROUP LNCTL=SDLC, DIAL=YES, ANSWER=ON, CALL=INOUT
LANLINE1 LINE
LANPU1  PU
```

Switched Major Node in Network Node 1

```
A1SWLAN  VBUILD TYPE=SWNET, MAXNO=8, MAXGRP=8
SWPU1A2A PU  IDBLK=001, IDNUM=00001, MAXPATH=5, MAXDATA=256, CPCP=YES,  *
          CPNAME=SSCP2A, PUTYPE=2, ADDR=C1, DISCNT=NO, NETID=NETA,      *
          LANSW=YES, MACADDR=400000092212, SAPADDR=4
PATH1A2A PATH DIALNO=400000092212, GRPNM=GP1A2A
ECHOA01  LU   LOCADDR=0
```

Switched Major Node in Network Node 2

```
A2SWLAN  VBUILD TYPE=SWNET, MAXNO=8, MAXGRP=8
SWPU2A1A PU  IDBLK=001, IDNUM=00001, MAXPATH=5, MAXDATA=256, CPCP=YES,  *
          CPNAME=SSCP1A, PUTYPE=2, ADDR=C1, DISCNT=NO, NETID=NETA,      *
          LANSW=YES, MACADDR=400000092211, SAPADDR=4
PATH2A1A PATH DIALNO=400000092211, GRPNM=GP2A1A
ECHOA02  LU   LOCADDR=0
```

Using ICA over an SDLC Connection between APPN Nodes

Figure 146 shows an ICA connection between two APPN nodes over an SDLC connection.

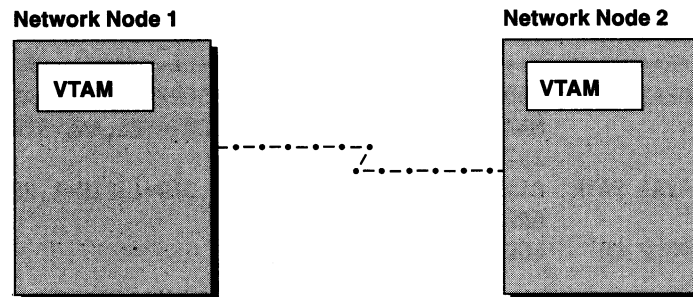


Figure 146. Using ICA over an SDLC Connection between APPN Nodes

Channel-Attachment Major Node in Network Node 1

ICASWVT1 VBUILD TYPE=CA

*

* NON-SWITCHED GROUP

GP1A2A1 GROUP LNCTL=SDLC,DIAL=NO,ISTATUS=INACTIVE

LN1A2A1 LINE ADDRESS=035,MODE=SEC,ISTATUS=INACTIVE

LS1A2A1 PU ADDR=C1,PUTYPE=2,XID=YES

NETAPPL1 LU LOCADDR=0,PACING=(1,1),VPACING=2,ISTATUS=INACTIVE

*

* SWITCHED GROUP

GP1A2A2 GROUP LNCTL=SDLC,DIAL=YES,ISTATUS=INACTIVE

LN1A2A2 LINE ADDRESS=037,CALL=INOUT,RETRIES=2,REPLYTO=5,AUTO=037

LS1A2A2 PU

Channel-Attachment Major Node in Network Node 2

ICASWVT2 VBUILD TYPE=CA

*

* NON-SWITCHED GROUP

GP2A1A1 GROUP LNCTL=SDLC,DIAL=NO,ISTATUS=INACTIVE

LN2A1A1 LINE ADDRESS=034,MODE=PRI,ISTATUS=INACTIVE

LS2A1A1 PU ADDR=C1,PUTYPE=2,XID=YES

*

* SWITCHED GROUP

GP2A1A2 GROUP LNCTL=SDLC,DIAL=YES,ISTATUS=INACTIVE

LN2A1A2 LINE ADDRESS=036,CALL=INOUT,RETRIES=2,REPLYTO=5,AUTO=036

LS2A1A2 PU

Switched Major Node in Network Node 1

```

SWND SVT1 VBUILD TYPE=SWNET,MAXNO=16,MAXGRP=16
*
*****
* DIAL IN/OUT GROUPS FOR VTAM TO VTAM AS PU_T21
*****
SWP1A2A2 PU      ADDR=C1,PUTYPE=2,CPNAME=SSCP2A,IDBLK=FAB,IDNUM=00001,      X
                MAXDATA=256,MAXPATH=1,DISCNT=YES,ANS=STOP,                X
                ISTATUS=ACTIVE
PTH1A2A2 PATH    DIALNO=ICASW21-9-111-0001,GID=1,PID=1,REDIAL=6,          X
                GRPNM=GP1A2A2
NETAPPL2 LU      LOCADDR=0,ISTATUS=ACTIVE

```

Switched Major Node in Network Node 2

```

SWND SVT2 VBUILD TYPE=SWNET,MAXNO=16,MAXGRP=16
*
*****
* DIAL IN/OUT GROUPS FOR VTAM TO VTAM AS PU_T2.1
*****
SWP2A1A2 PU      ADDR=C1,PUTYPE=2,CPNAME=SSCP1A,IDBLK=FAB,IDNUM=00001,      X
                MAXDATA=256,MAXPATH=2,DISCNT=YES,ANS=STOP,                X
                ISTATUS=ACTIVE
PTH2A1A2 PATH    DIALNO=ICASW21-9-222-0001,GID=1,PID=1,REDIAL=YES,          X
                GRPNM=GP2A1A2
APPL2      LU      LOCADDR=0,ISTATUS=INACTIVE

```

Defining a Network Node Server List

A network node server list indicates the network nodes that can act as servers for an end node. It is not required. An end node becomes aware of adjacent network nodes capable of acting as its network node server through link activation or system definition. However, without a network node server list, the adjacent network node chosen as an end node's server is directly related to when the end node becomes aware of that particular network node, relative to other adjacent network nodes. Without a network node server list, an end node establishes a CP-CP session with the first such network node that it becomes aware of, and this network node acts as the end node's server. There is no predictability in this scheme; it is all timing dependent.

You may want to control which network node is selected by an end node. For example, you may want to exclude a particular network node from network node server responsibilities; acting as a server does involve some overhead, such as originating search requests or executing domain broadcasts. You may also want to isolate particular end nodes from certain network nodes for security reasons.

To restrict the network node server to one that has the same level of support for SLU-initiated sessions as the end node, you can use the SLUINIT operand on the NETSRVR definition statement. For example, to require that a network node server be able to support SLU-initiated sessions, you can define SLUINIT=REQ. The SLUINIT=REQ specifies that CP-CP sessions can only be established with a network node that supports SLU-initiated sessions. SLUINIT=REQUIRED is the default. If you define SLUINIT=OPT, then CP-CP sessions are established with a network node server regardless of whether it supports SLU-initiated sessions.

You can create a prioritized network node server list by allowing the ORDER operand to default to FIRST on the NETSRVR definition statement. ORDER=FIRST causes the end node to always start at the top of the list when searching for a network node server. If ORDER=NEXT, the end node selects the network node in the list following the last node selected. ORDER=NEXT gives no preference to any node on the list.

Creating a Network Node Server List

To create a network node server list for an end node, create a VTAMLST member containing a VBUILD TYPE=NETSRVR definition statement, and one or more NETSRVR definition statements for each network node that you want in that network node server list. You can also include a nameless entry to allow the end node to select any other known adjacent network node that meets the defined criteria as its network node server. A sample network node server list follows:

```

          VBUILD TYPE=NETSRVR
CP1A    NETSRVR  NETID=NETA
CP2A    NETSRVR  NETID=NETA
CP3A    NETSRVR  NETID=NETA
          NETSRVR

```

Note: Any network node named on a NETSRVR definition statement previously considered and rejected as a network node server is not retried during nameless entry processing.

Activating, Replacing, and Displaying a Network Node Server List

You can activate a network node server list at startup by specifying the appropriate VTAMLST member name in your configuration list, or by using the VARY ACT command at an end node. If you use the VARY ACT command to activate the network node server list after a CP-CP session has already been established, the list is not used until the existing CP-CP session between the end node and its current network node server is lost or terminated. Use the VARY TERM command at the end node to deactivate the CP-CP session between the end node and the network node acting as the end node's server. VTAM will then use the network node server list in selecting a new server for the end node.

You can also dynamically replace an existing network node server list. Create a new list and, as described above, use the VARY ACT command to activate it and the VARY TERM command to force VTAM to use the new list.

Use the DISPLAY NETSRVR command at an end node to display the end node's network node server list, the end node's current network node server, and other network nodes known to the end node.

APPN Multiple Network Connectivity

APPN multiple network connectivity support enables connectivity between APPN networks, without the exchange of topology information between the networks. APPN multiple network connectivity support may exist between APPN networks that have different network IDs, and also between APPN subnetworks that have the same network ID.

APPN multiple network connectivity support utilizes border nodes at the boundaries between APPN networks or subnetworks. With APPN multiple network connectivity

support, full APPN directory and session initiation support is available across these boundaries. APPN directory search activity can be managed through several user-definable parameters for border nodes. Class-of-service definitions do not need to be consistent across network or subnetwork boundaries, allowing for independent management.

APPN multiple network connectivity support enables you to eliminate topology exchange between APPN networks or subnetworks when the exchange of such data is undesirable (for example, for networks of different enterprises). Topology exchange is eliminated between border nodes. This way, you can protect smaller subnetworks from the volume of topology in a larger subnetwork or from processing searches for resources that do not exist in the smaller subnetwork. Thus, APPN multiple network connectivity support can be used to manage the increase in topology and directory search activity as an APPN network grows.

For example, the APPN network in Figure 147 on page 547 has been divided into four subnetworks. With APPN multiple network connectivity support, full APPN directory and session initiation support is available across all subnetwork boundaries, yet the creation of smaller subnetworks protects each from the volume of topology and search processing that would exist in the entire APPN network (which could, of course, extend beyond what is shown).

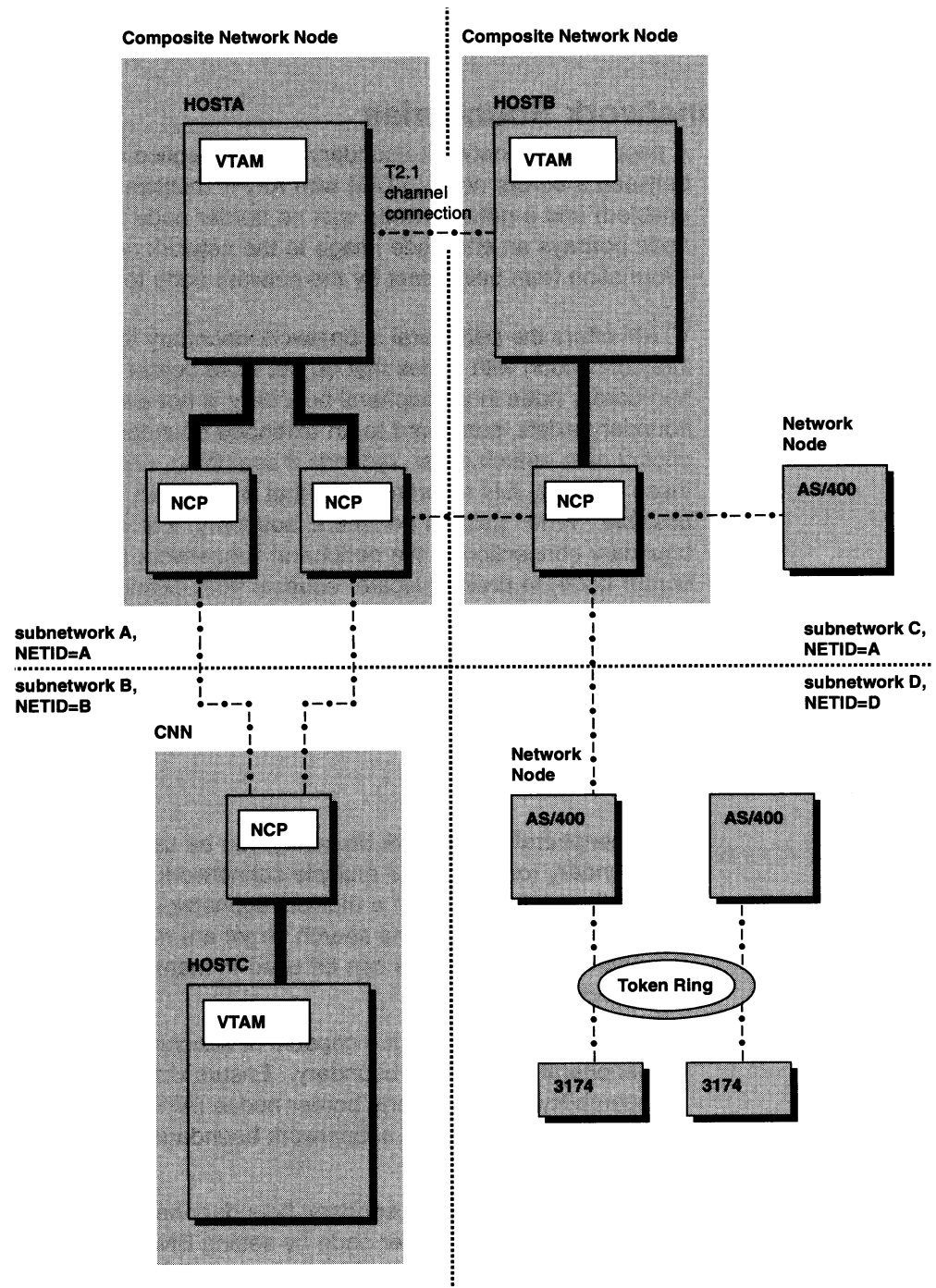


Figure 147. APPN Subnetworks through APPN Multiple Network Connectivity Support

There are two types of border node boundaries:

- Peripheral subnetwork boundaries
- Extended subnetwork boundaries.

Both of these types of border node boundaries are discussed below. In addition, Appendix I, "Border Node Connection Types" on page 691, lists connection types for various VTAM and partner node combinations.

Note: APPN multiple network connectivity support cannot be used to connect two APPN networks or subnetworks through a connection network.

Peripheral Subnetwork Boundaries

A peripheral subnetwork boundary is a connection over a subnetwork boundary between a border node (VTAM with APPN multiple network connectivity support enabled) and a network node with no border node function. The VTAM border node portrays an end node image to the network node, thus preventing topology information from being sent by the network node to VTAM.

VTAM offers the peripheral subnetwork boundary to provide APPN subnetwork interconnection with nodes that do not have border node function. Because the non-border node in a peripheral boundary is not aware that the subnetwork boundary exists, compared to an extended boundary, there is considerable loss of control over search order, redundant searching, and looping search prevention. For these reasons, it is recommended that you use an extended boundary whenever possible. When using a peripheral boundary, it is wise to use a single subnetwork boundary connection to the peripheral subnetwork (the subnetwork of the non-border node) to prevent locate requests from exiting the subnetwork over one boundary and then re-entering over another boundary. If a peripheral subnetwork has more than one boundary, the user should customize routing via a user exit or routing definitions to ensure that the requests do not re-enter the same peripheral subnetwork. This precaution is necessary to avoid possible failed sessions due to a looping locate path.

Notes:

1. A peripheral subnetwork boundary can be used as either the first or last boundary (or both) on a multiple subnetwork search path. If there is an intermediate subnetwork on a multiple subnetwork search path (that is, neither the search originator nor the search target are members of that subnetwork), only extended border nodes can be used to manage the intermediate subnetwork's boundaries.
2. Not all network nodes are capable of attaching to a VTAM border node via a peripheral subnetwork boundary. Ensure the product and release support interoperability with peripheral border nodes (APPN option set 1013) before attempting to establish subnetwork boundaries between VTAM and a network node.
3. It is possible to get a peripheral boundary between two VTAMs that both support extended border node by setting BN=NO in one of the two nodes. However, this is strongly discouraged, as it bypasses the valuable looping and redundant search controls that are present if the connection is an extended subnetwork boundary.

Extended Subnetwork Boundaries

An extended subnetwork boundary is a connection between two border nodes that support extended subnetwork boundaries. Although topology flows are still restricted across the boundary, both nodes realize that a network node exists on the other side, allowing for additional subnetwork hops and increased control over routing.

Note: Some border nodes do not support extended subnetwork boundaries. If one of two communicating border nodes does not support extended subnetwork boundaries, a peripheral subnetwork boundary is used.

Using APPN Multiple Network Connectivity Support

To enable APPN multiple network connectivity support, use the BN=YES start option at a VTAM network node (that is, NODETYPE=NN must also be specified). BN=YES indicates that VTAM is capable of establishing subnetwork boundaries.

Note: NCP Version 7 Release 1 or later is required for both extended and peripheral subnetwork boundaries through an NCP.

You need to code only the BN=YES start option to take advantage of APPN multiple network connectivity support. Some controls are available for customization:

- An adjacent cluster routing definition list can be used to customize routing between subnetworks. Start with the VBUILD TYPE=ADJCLUST definition statement.
- The BNDYN start option is used to define how VTAM adds entries to the adjacent cluster routing table. By default, VTAM adds a limited set of nodes to the table.
- The BNORD start option is used to control the search order when searching across subnetwork boundaries. By default, VTAM gives preference to nodes for which the most recent search was successful and nodes whose network ID matches that of the DLU. Specify BNORD=DEFINED to use the defined search order.
- The SNVC start option is a number between 1–255 that specifies the maximum number of subnetworks to which a request can be forwarded, including this subnetwork. For example, the default for the SNVC start option is 3, indicating that a request can be routed from this subnetwork across up to two subnetwork boundaries, for a total of three subnetworks visited when you count the original subnetwork. Given the connectivity shown in Figure 147 on page 547, the default would not allow the routing of a request from subnetwork D to subnetwork B (SNVC=4 is needed).
- Class of service mapping definitions can be placed in a COS mapping table to:
 - Map non-native COS definitions to native COS definitions
 - Map native COS definitions to non-native COS definitions.

Start with the VBUILD TYPE=BNCOSMAP definition statement.

- The APPNCOS start option is used to specify the default COS used for requests received over a subnetwork boundary with an unrecognizable COS listed. The value for APPNCOS is used only after COS mapping is attempted. The default value is NONE.
- The CACHETI start option defines the number of seconds that routing information about a previous locate search is stored, if applicable. The default is eight minutes.
- The NATIVE operand is specified on the ADJCP definition statement in the adjacent CP major node, or on a PU definition statement defining an APPN connection. The NATIVE operand is valid only when BN=YES is specified. Code NATIVE=NO to define a subnetwork boundary between this node and the named adjacent CP, or between this node and the CP represented by the PU

statement. Use `NATIVE=NO` when both nodes have the same network ID, but a subnetwork boundary is desired. For example, in Figure 147 on page 547, `NATIVE=NO` was used to create the subnetwork boundary between subnetwork A and subnetwork C. The `NATIVE` operand is required on only one side of a network or subnetwork boundary, though it is allowed on both sides as long as they do not conflict.

Network Routing and Resource Location for APPN Nodes

VTAM uses topology and routing services (TRS) and directory services (DS) to learn and maintain information on the topology and resources of the network.

TRS maintains information about nodes, transmission groups (TGs), and classes of service so that appropriate routes through an APPN network can be calculated.

TRS creates a network topology database, in which it maintains information about the current topology of the intermediate routing portion of an APPN network (network nodes and the transmission groups interconnecting them). This network topology database is replicated at every network node in an APPN network. Whenever a new network node or TG is added, or whenever the characteristics of an old resource change, a topology database update (TDU) that describes the current characteristics of the resource is generated and propagated to all network nodes.

When a network node first joins an APPN network (that is, establishes CP-CP sessions with other nodes), the other nodes send the joining node a copy of the network topology database. This is known as topology exchange. (A topology exchange that occurs when a network node first joins an APPN network is also called a full topology exchange or an initial topology exchange.) If the joining node later experiences planned or unplanned down time, it can relearn the network topology by receiving all the topology updates that were exchanged since its topology database was last updated.

Directory services maintains a database that contains information on the location of LUs in the network. Entries are added to this database by resource definition, resource registration, and dynamically through searches. This means even if a session is requested to an LU whose location is not currently stored in the database, the location of this LU can be learned, stored in the database, and session establishment will occur.

When a session is requested with an LU, VTAM uses its search capability and the directory services database to determine what node the resource is on. The search results give the location of the resource but not the route. VTAM dynamically determines the best route for the session using the APPN class of service requested for the session and the characteristics of the intermediate nodes and links stored in the topology database.

The defaults for the definition of the node and link characteristics and VTAM's ability to dynamically update the contents of the databases allow sessions to be established without requiring any definitions for resources outside of the domain of that VTAM.

There are techniques you can use to improve the efficiency of resource location and session setup. These techniques include coding a central directory server and registering resources to a central directory server or a network node server.

Resource registration occurs by default for some resources, but you can modify which resources are registered.

You can also checkpoint the topology and directory databases to ensure that information is retained during system outages. Checkpointing the directory database increases the chances that a resource can be located with a directed search instead of a broadcast search. Directed and broadcast searches are described in “Types of Searches.” Checkpointing the topology database reduces the number of topology database updates that must flow at startup.

You may also need to adapt the class of service definitions, link and node characteristics to your network; such as coding some links as secure and requesting a secure class of service for certain sessions.

The following sections give information on searches, checkpointing, and APPN class of service.

Types of Searches

In order to establish a session, the location of the partner resource must be known. If the location of the partner resource is not statically defined, it can be learned dynamically as a result of a directed, broadcast, or cross-subnetwork search. A directed search is used when VTAM has location information for the resource as the result of a previous search or if there is a central directory server in the network. A broadcast search is used when VTAM has no information on the location of the resource. A cross-subnetwork search is used to determine the location of a partner resource in another subnetwork.

Directed Search

What is a directed search? A directed search uses information stored in the directory services database of a network node server to direct the search to the location of the requested LU. A directed search is sent to the node recorded as the owner of the requested LU to verify the information. A directed search can be sent to a network node server from an end node and to an end node from a network node server. A directed search can also be sent from a network node to a central directory server when the network node does not have information on the location of the destination logical unit or following a failed search.

Who originates a directed search? A directed search is originated by a network node server for the LUs it serves, by a central directory server as a result of a search request from a network node, or by an end node to its network node server.

When is a directed search needed? A directed search is needed when a network node server or a central directory server receives a request for an LU for which it has a location stored in its database. The search is used to verify the information. A directed search is also used by a network node server to send a search request to a central directory server if the network node server has no information on the location of a requested resource. A directed search is used by an end node to request location information about a resource from the end node's network node server.

Broadcast Search

What is a broadcast search? A broadcast search does not use database information about the location of a requested LU to propagate the search, because a directed search using database information has failed or there is no database information for the requested resource. Each network node receiving the search request sends the search to each of its adjacent network nodes; this allows the whole network to be searched. (VTAM end nodes do not allow searches for resources that are not registered. With other types of end nodes, broadcast searches include all domain end nodes that have requested that they be searched for the specific resource type.)

When the search reaches the network node serving the destination resource, that node sends back a positive reply to the first search request it receives. All subsequent search requests received by that network node as part of the same search are answered with a negative reply.

Who originates a broadcast search? A broadcast search is initiated only by a network node server for the LUs it serves if there is no central directory server defined in the network. If there is a central directory server defined, only the central directory server can issue a broadcast search. If a central directory server is active in your network, a network node server sends a directed search to the central directory server instead of issuing a broadcast search. Only if the directed search is unable to reach the central directory server does the network node server issue its own broadcast search.

When is a broadcast search needed? A broadcast search is needed when there is no directory information giving the location of a requested LU or a directed search using existing directory information fails.

- **Broadcast searches in a network without a central directory server**

In a network without a central directory server, broadcast searches are performed by any of the network nodes. These network nodes may be acting on behalf of LUs attached directly to them, LEN-attached LUs, or (when the network node is acting as a network node server) on behalf of LUs attached to end nodes that they serve. When the network node receives a request from an LU for a session with another LU, it checks for information in the database about the location of the partner LU. If it does not find the location information for the partner LU, the network node server of the requesting logical unit initiates a broadcast search.

The location information of the resource is not communicated to the other network nodes. If another network node server receives a request for the same resource and does not have the location information, it issues a broadcast search.

- **Broadcast searches in a network with a central directory server**

In a network with a central directory server, broadcast searches are performed only by the central directory server. If a network node server does not know the location of a partner LU, it sends a directed search to the central directory server instead of issuing a broadcast search. The central directory server checks its database and issues a broadcast search only if it does not have the information. When the requested LU is located, the central directory server updates its directory and returns the information to the requesting network node. The network node then updates its database with the information and initiates the session with the partner LU. After a resource's location is stored in

the central directory database, no further broadcast searches are required, until the resource moves or is deactivated.

Cross-Subnetwork Search

What is a cross-subnetwork search?: A cross-subnetwork search is designed to find an LU outside the local APPN subnetwork. Since other APPN searches are sent between nodes that are in the same APPN subnetwork, this search type is employed at border nodes to forward searches for a requested LU to other APPN subnetworks. The target nodes of a cross-subnetwork search may be found in the directory services database, learned dynamically, or explicitly defined via definition statements.

Who originates a cross-subnetwork search? The first border node encountered as the search leaves the requester is responsible for originating a cross-subnetwork search. The border node forwards the request over locally managed subnetwork boundaries. Other nodes in non-native subnetworks then perform their search logic to find the resource in their local subnetwork or across their subnetwork boundaries. These searches are performed sequentially until the target resource is located.

Is a cross-subnetwork search simply a broadcast search sent over subnetwork boundaries?: No. Cross-subnetwork search logic finds resources across subnetwork boundaries with minimum disturbance to other subnetworks. The sequential fashion in which the searches are sent allows the search to terminate as soon as the resource is located, without having to search every connected subnetwork. In addition, routing information can be both defined or learned dynamically, which directs the search toward the subnetwork that is most likely to own the resource.

Minimizing Broadcast Searches to Improve Performance

Two techniques for controlling searches are defining a central directory server to minimize the number of broadcast searches and registering resources to a directory database.

Also, an installation-wide directory services management exit provides control over the extent to which VTAM conducts search requests in the network in which the directory services management exit is loaded. The directory services management exit is the interface between VTAM's directory services and user-written code. VTAM conducts searches based on steps authorized by the user code. For more information on the directory services management exit, see "Directory Services Management Exit Routine" in *VTAM Customization*.

Central Directory Servers

What is a central directory server? A central directory server is a focal point for broadcast searches in the network. It uses broadcast and directed searches to locate resources. When a central directory server is active in the network, all network node servers issue directed searches to the central directory server instead of issuing a broadcast search.

What are the benefits of a central directory server? A central directory server reduces the number of broadcast searches in your network. When the central directory server performs a broadcast search for a network node, it stores the location of the requested LU in its database. If another network node server sends

a directed search for the same resource to the central directory server, the central directory server does not have to issue another broadcast search because it already has the resource in its database.

Because a central directory server can receive search requests from many network nodes, its database grows faster and contains more resources than that of a network node server, which only receives the search requests of its served LUs. This centralization increases the chance that a resource will be found in the database and a broadcast search will not be required.

How does a central directory server handle a search request?: When the central directory server receives a search request, it checks its database for the resource.

If it does not find the resource in its database and there are other central directory servers in the network, it sends the search to the other central directory servers only. If the central directory server receives a positive reply from any of the other central directory servers, it updates its own database with the information and notifies the originating network node of the location of the requested LU. If the central directory server receives negative replies from all the other central directory servers, it initiates a broadcast search. If the central directory server that received a search request locates the resource in its own database, it verifies the information with a directed search and sends a reply to the originating network node server with the location of the requested LU.

If the central directory server does not find the resource in its database and there are no other central directory servers in the network, it initiates a broadcast search.

What do you need to define to take advantage of a central directory server? Use the CDSERVER start option to define VTAM as a central directory server. You do not need to define anything on the other network nodes. The other network nodes will find out about the existence of the central directory server through normal topology exchanges.

Registering Resources

Resource registration places information about the location of resources in a directory services database. This registration reduces broadcast searches by ensuring that a resource will be found in the directory services database. Resources can be registered to a directory database on a network node server and/or to a central directory server. By registering resources to a network node server, you ensure that the resources will be found during a search. By registering resources to a central directory server, you ensure that the resources will be found without requiring a broadcast search. The defaults for registration change depending on the type of resource and whether the resource is located on an end node or a network node.

Because the characteristics of an independent LU and of the node on which it resides are not recorded by VTAM, it is difficult to determine the appropriate level of registration for all cases. Therefore, independent LUs are not registered by default.

VTAM end nodes register their dependent LUs with their network node server by default. This includes all resources defined by LOCAL definition statements and LU definitions with a non-zero LOCADDR value. A resource on a VTAM end node that

is not registered to the network node server will not be found during a search unless that resource has initiated a request through the network node server. Registering all end node resources insures that they will be located by a search. You can also specify that these resources be registered to a central directory server.

Applications are registered to a central directory server by default, with the exception of TSO applications. For more information on default registration of TSO applications, see “Resource Registration in an APPN Network” on page 614.

Note: If a resource is never going to be a destination resource, there is no need to register it.

Resources on a network node can also be registered to a central directory server if one exists. By default, network nodes register only their applications to a central directory server.

Controlling the Timeliness and Size of the Directory Database

By using the start options DIRTIME and DIRSIZE, you can control how long a resource remains in the directory services database and how many dynamic entries are maintained.

The DIRTIME start option specifies the amount of time a resource remains in the directory database without being used before it is deleted. The default value for this start option is eight days.

The DIRSIZE start option defines the maximum number of dynamically added resources that are stored in the database. After the number of dynamic resources reaches the maximum, new resources are added to the database in place of the resources that have gone the longest without being referenced.

Checkpointing of the TRS Database and the Directory Database

When a network node is added to an APPN network, it must complete a full topology exchange with the other nodes in the network. In large APPN networks, full topology exchanges can involve the transfer of large amounts of data, affecting performance by slowing down the CP-CP session activation process.

When a node that has just joined the APPN network attempts to establish sessions, it has to learn the location of all resources that it wishes to establish sessions with by issuing a broadcast search or a directed search to a central directory server.

By checkpointing the TRS and directory services databases, the topology and resource location information can be saved and used if VTAM is restarted.

Checkpointing saves the databases to the checkpoint data sets at the time either of the following occurs:

- The VTAM operator issues a MODIFY CHKPT command.
- The VTAM operator issues a HALT or HALT QUICK command to terminate the VTAM network node.

This process of saving the databases is also called database hardening.

Note: The databases are not saved if the VTAM operator:

- **MVS, VM** Issues a HALT CANCEL command to terminate the VTAM network node

- **VSE** Cancels VTAM by canceling VTAM's partition.

When a database is checkpointed, it is written to a checkpoint data set. This data set is read by VTAM when it is restarted and the information is placed in the databases. If checkpointing is not used, both the topology and directory services databases have to be completely rebuilt. With checkpointing only those changes to resources and topology since the last checkpoint need to be learned.

To specify which of the checkpointed databases are to be loaded at startup time, you can use the INITDB start option. By default, both the topology and directory services databases are loaded. You can also specify topology only, directory services only, or neither.

APPN Class of Service

What is APPN class of service?

APPN class of service defines the characteristics of a route for a session. Unlike subarea class of service, which defines the actual routes to be used for a session, APPN class of service defines the characteristics of the nodes and transmission groups (TGs) to be used for a session. VTAM chooses an APPN route by matching the required characteristics defined in the COS definition with the actual characteristics of the nodes and TGs in the topology database.

What information is specified in the APPN class of service?

The APPN class of service indicates the desired characteristics of the nodes and lines in a route and the transmission priority of sessions. The possible line characteristics that can be specified include the security, speed (line capacity), propagation delay, and cost (both per unit of time and per byte). There are also three user-defined parameters that can be used to specify additional characteristics. The node characteristics that can be specified are the level of congestion allowed when selecting a node for routing and the desirability of the node for intermediate routing. The desirability of using a node for intermediate routing is called route resistance.

The characteristics are specified in ranges, with a low weight assigned to the most desirable set of ranges and a high weight assigned to the least desirable set of ranges. The first row in the default class of service has a very low weight and has the most restrictive set of ranges. The next set of ranges is less restrictive but has a higher weight. (These sets of ranges are called LINEROWs and NODEROWs.) The last set of ranges has the widest range of acceptable characteristics but has a very high weight. By coding the rows in this manner, the best resources (lowest weight and best characteristics) are chosen when determining the route.

Do you have to code anything to make APPN class of service work?

IBM supplies seven default APPN class of service definitions. The TG and node characteristics also default. For APPN-only sessions, VTAM locates an available route without any additional coding.

What are the IBM-supplied default classes of service?

The following APPN COS definitions are supplied in the COSAPPN member or file of the **MVS** SYS1.VTAMLST library, **VM** LOCALSAM disk, or the **VSE** VTAM production sublibrary:

CPSVCMG	For CP-CP sessions. Provides connectivity at network transmission priority.
SNASVCMG	For LU-LU CNOS sessions. Provides connectivity at network transmission priority.
#BATCH	A general batch-oriented class of service that uses low transmission priority. High bandwidth and low cost are considered more important than short delay.
#BATCHSC	Same as #BATCH, except that a minimum level of security is required.
#INTER	A general, interactive-oriented class of service that uses high transmission priority. Short delay is considered to be more important than high bandwidth and low cost.
#INTERSC	Same as #INTER, except that a minimum level of security is required.
#CONNECT	For LU-LU sessions. Provides connectivity at medium transmission priority.

What do you get if you take all the defaults?

The name of the APPN class of service definition associated with the mode is specified by using the APPNCOS operand on the MODEENT macroinstruction. Values are supplied for the APPNCOS operands in the IBM-supplied logon mode table (ISTINCLM). The value for APPNCOS in the IBM-supplied logon mode table is coded with the APPN class of service that most closely matches the type of session the logon mode entry is intended for. For example, if the logon mode name is coded as BATCH, BAT13790, BAT23790, or #BATCH, then the APPNCOS operand value is #BATCH¹. For the complete list of logon mode names and APPNCOS values, see Table 29.

Table 29 (Page 1 of 2). ISTINCLM APPNCOS Values Based on LOGMODE Operand Values.

IBM-Supplied Mode Name	APPNCOS Value
CPSVCMG	CPSVCMG
CPSVRMGR	SNASVCMG
SNASVCMG	SNASVCMG
#BATCH BATCH BAT13790 BAT23790	#BATCH
#BATCHSC	#BATCHSC
#INTER INTERACT INTRACT INTRUSER	#INTER
#INTERSC	#INTERSC
DSILGMOD DSC4K D32901 EMUDPCX SCS xxx3270x xxx3277x xxx3278x xxx3790x etc.	#CONNECT

¹ # represents X'7B'. You should use whichever character on your system equates to X'7B'.

Table 29 (Page 2 of 2). ISTINCLM APPNCOS Values Based on LOGMODE Operand Values.

IBM-Supplied Mode Name	APPNCOS Value
ISTCOSDF	#CONNECT
IBMRDB	#CONNECT

What is the APPN class of service default in a user-coded logon mode table?

If the APPNCOS operand is not coded on the MODEENT macroinstruction, VTAM checks the COS operand value.

- If the COS operand value is coded, VTAM checks for an APPNCOS definition by the same name as the COS name. If VTAM does not find an APPNCOS definition by the same name as the COS, it fails the session request, unless the APPNCOS start option is defined with a substitute APPN class-of-service name.
- If the COS operand is not coded, VTAM defaults APPNCOS to #CONNECT, regardless of the setting for the APPNCOS start option.

How do I add APPN class of service to my logon mode table?

You can add APPN class of service to your user-coded logon mode table in one of two ways:

- You can code the APPNCOS operand for each logon mode entry in the table.
- You can create new APPNCOS definitions with the same name as the subarea COS.

To create APPNCOS definitions with the same name as the subarea COS, you can copy the default definition for APPN class of service that best fits that logon mode and change the name to match the COS operand coded on that logon mode.

How does VTAM use the class of service to choose a route?

VTAM chooses a route by comparing the actual characteristics of the available nodes and TGs to the allowed characteristic ranges specified in the requested class of service. For each APPN class of service entry, there are one to eight LINEROW operands and one to eight NODEROW operands. These operands give up to eight acceptable sets of characteristics for the lines and nodes in each class of service. The WEIGHT parameter on the NODEROW and LINEROW operands is coded to indicate the desirability of that set of characteristics. The lower the value of the WEIGHT parameter the higher the desirability of a node or TG that fits that set of characteristics.

For each of the LINEROW and NODEROW operands, an acceptable range is coded. For each potential line or node in a route, VTAM compares the node or TG characteristics to the LINEROW or NODEROW operand values in the requested class of service. For an entire route to be acceptable for a given class of service all nodes and TGs for that route must be acceptable for the requested class of service.

If more than one route fits the required characteristics of the class of service, VTAM chooses the one with the total least weight, and therefore the most desirable characteristics. If no route matches the required characteristics of the class of service,

the session fails; for example, if the class of service requires a secure route and VTAM cannot find a complete route using only secure TGs.

Influencing Session Routing

The RESUSAGE start option is used to affect the number of times a given resource is used in creating a route. RESUSAGE specifies the number of times a resource can be used before it is considered overused. Once a resource is overused, any routing trees using it are reconstructed in an attempt to use alternate resources.

Example of Weight Computation

The following example illustrates how a class of service is used to assign a weight to an element of a route. This example illustrates what weight is assigned to a TG with default characteristics using the #CONNECT class of service. Default LINEROW values for the #CONNECT class of service are shown in Table 31. Default characteristics of a TG are shown in Table 30.

Table 30. Default TG Characteristics

TG number	CAPACITY	COSTBYTE	COSTTIME	PDELAY	SECURITY	UPARM1 UPARM2 UPARM3
TG1	8K	0	0	TERRESTR	UNSECURE	128 128 128

Table 31. #CONNECT Class of Service LINEROW Values

LINEROW	CAPACITY	COSTBYTE	COSTTIME	PDELAY	SECURITY	UPARM1 UPARM2 UPARM3	Weight
1	4M MAXIMUM	0 0	0 0	MINIMUM NEGLIGIB	UNSECURE MAXIMUM	0 255	30
2	56000 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	60
3	19200 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	90
4	9600 MAXIMUM	0 0	0 0	MINIMUM TERRESTR	UNSECURE MAXIMUM	0 255	120
5	19200 MAXIMUM	0 0	0 0	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	150
6	9600 MAXIMUM	0 128	0 128	MINIMUM PACKET	UNSECURE MAXIMUM	0 255	180
7	4800 MAXIMUM	0 196	0 196	MINIMUM MAXIMUM	UNSECURE MAXIMUM	0 255	210
8	MINIMUM MAXIMUM	0 255	0 255	MINIMUM MAXIMUM	UNSECURE MAXIMUM	0 255	240

The values shown in Table 31 for capacity, cost per byte, cost per unit of time, propagation delay, security level, and the user-defined characteristics (UPARM1, UPARM2, and UPARM3) represent ranges, with the top value in a LINEROW representing the minimum value and the bottom value in a LINEROW representing

the maximum value. In comparing these values with the TG characteristics shown in Table 30, the first match occurs at row 7.

The cost per byte, cost per unit of time, security, and user-defined characteristics are acceptable for all rows. However, the possible LINEROW values for capacity (number of bits per second that the link can transmit data) are MINIMUM, 600, 1200, 2400, 4800, 9600, 14400, 19200, 48000, 56000, 64000, 4M, 16M, and MAXIMUM, and effective capacity is not within the range for rows 1 through 6. The possible values of propagation delay (from lowest to highest) are:

- MINIMUM
- NEGLIGIB
- TERRESTR
- PACKET
- LONG
- MAXIMUM.

Propagation delay is not within the range for the first row.

So, the default values for a TG match both rows 7 and 8 of the #CONNECT class of service. Row 7 has a weight of 210 and this weight is assigned to this TG for the route being calculated. Row 8 has a weight of 240, but even if it had a lesser weight it would not be used. (It is recommended that you code your rows from least weight to highest weight to ensure that the least weight row is used in the event of multiple matches.)

This TG is used if no other TG is available. If another TG is available, it is used if its weight is less than 210. In the event two or more TGs are equally weighted, one is chosen at random.

If this TG is being considered for part of a route that involves nodes and other TGs, the weight of this TG would be added to the weights of the other TGs and nodes, to produce a total weight for the route.

If the origin or destination of this TG is a virtual node, the weight of the TG is divided by two and the calculated value is used in route calculation.

For an example of how to use the node and TG characteristics to force a route, see Appendix H, "Forcing an APPN Route in a VTAM Network" on page 685.

Verifying a VTAM APPN Network

To verify that your VTAM APPN network functions properly, follow these steps:

1. Start VTAMs with appropriate APPN start options.
2. Activate NCP major node, or XCA major node for 3172 connection.
3. Activate CP-CP session-capable APPN connections (for example, lines and PUs) to adjacent nodes and verify that CP-CP sessions are automatically started. The following message is issued:

```
IST1096I CP-CP SESSIONS WITH adjcpname ACTIVATED
```

4. Display adjacent control points. For example, if VTAM was started with DYNADJCP=YES, issue:

```
D NET,ID=ISTADJCP,E.
```

5. From VTAM network nodes, display the topology of the APPN network. For example:

```
D NET,TOPO,ID=cp_name,LIST=ALL.
```

6. From VTAM end nodes, display the network node server list. For example:

```
D NET,NETSRVR,E.
```

7. Terminate one of the CP-CP sessions. For example, issue:

```
V NET,TERM,LU1=cp_name,LU2=adjcp_name.
```

8. Re-establish the CP-CP session. For example, issue:

```
V NET,ACT,ID=adjcp_name,IDTYPE=CP.
```

9. To verify end node registration, first activate a resource defined with REGISTER=NETSRVR on the end node (note that application program minor nodes default to REGISTER=NETSRVR). Then display the resource from the network node server's directory database. For example, issue:

```
D NET,DIRECTRY,ID=resource_name.
```

10. Verify sessions by exchanging requests between APPN nodes.

11. Halt VTAM. Note that the HALT or HALT,QUICK commands checkpoint the topology and routing services database.

Part 5. Tuning VTAM for Your Environment

Chapter 11. Tuning VTAM for Your Environment	565
Introduction to Tuning	565
Tuning Tools	566
Estimating Active Sessions	566
Common Storage Areas	566
Buffer Pools	566
NCP Slowdown	567
Coattailing	567
Tuning Tools	567
Monitoring Common Storage Areas	567
Analyzing Slowdown	569
Gathering Tuning Statistics	569
Analyzing Tuning Statistics	576
Determining the Amount of Coattailing in Your System	577
Migrating from User-Replaceable Constants	577
Estimating the Number of Active Sessions	577
EAS Operand for Application Programs	578
EAS Operand for Independent Logical Units	578
Common Storage Areas	578
Common Storage Areas for MVS and VM	578
Common Storage Areas for VSE	579
DISPLAY STORUSE Pools	579
Buffer Pools	583
Types of Buffer Pools	583
Buffer Pool Allocation	585
HOT I/O Detection/Termination	592
Controlling NCP Slowdown	594
Maximizing Coattailing	595
Controlling Outbound Coattailing	595
Controlling Inbound Coattailing	596
Coattailing for SNA Controllers	597
Coattailing for Channel-to-Channel Operations	603

Chapter 11. Tuning VTAM for Your Environment

This chapter describes the task of tuning VTAM. It contains the following sections:

- Introduction to Tuning
- Tuning Tools
- Estimating the Number of Active Sessions
- Common Storage Areas
- Buffer Pools
- Controlling NCP Slowdown
- Maximizing Coattailing.

Introduction to Tuning

Tuning is the process of balancing the network load among resources to eliminate congestion in any one resource. You need to tune VTAM to give you optimal service while using the least amount of resources. The objectives of tuning are to use storage in the host processor and in the communication controller more efficiently and to lessen the load on the host processor. For example, to use storage more efficiently, you should avoid allocating too many VTAM buffers in the host and choose a more appropriate buffer size in the communication controller.

The elements of a network can be regarded as a series of related storage spaces. A host processor provides storage for VTAM and application programs. Communication controllers contain storage for NCPs and related programs. Cluster controllers and programmable peripheral nodes also contain storage space.

When VTAM supports telecommunication services for your system, VTAM must use storage to build control blocks to keep track of sessions, blocks of data, and other information. If your system needs exceed your storage capacity, then you might experience degraded response times. Similarly, an NCP can exceed its storage capacity if it receives more data from other parts of the network than it can send out.

VTAM uses storage for the following:

- VTAM modules
- VTAM buffer pools
- Tables representing major and minor nodes
- Tables representing routing capabilities of the SSCP
- Tables representing sessions controlled by the SSCP
- Temporary work space used for VTAM operator commands and session establishment or termination requests.

The amount of storage required for tables and buffer pools depends on the number of major nodes being defined and the number of sessions required by application programs. By increasing VTAM's efficiency, you can free storage for other uses.

Tuning Tools

To tune your environment, you need to be able to collect information about your network. You need to know how to gather and analyze tuning statistics, determine the amount of coattailing that is taking place in your network, analyze slowdown conditions, and monitor your common storage areas.

VTAM provides operator commands and other facilities to help you gather this information. See “Tuning Tools” on page 567 for a description of the tools that you can use to gather data to tune your network.

Estimating Active Sessions

When VTAM provides session services for applications and logical units, VTAM allocates storage to handle information about where data is to be sent. You can help system performance by estimating the number of sessions that VTAM will have with a particular application program or logical unit. This way, VTAM can set aside an appropriate amount of space for handling sessions. For further information, see “Estimating the Number of Active Sessions” on page 577.

Common Storage Areas

VTAM uses **MVS** Common Service Area, **VM** GCS common segment, and **VSE** System GETVIS Area to maintain some buffer pools and control blocks. Common storage areas are also used for other system needs, so you need to monitor VTAM's use of common storage areas (for information see “Monitoring Common Storage Areas” on page 567) and you might want to control VTAM's use. For more information on limiting VTAM's use of common storage areas, see “Common Storage Areas” on page 578.

Buffer Pools

VTAM uses buffer pools to control the handling of data: VTAM control blocks, I/O buffers, and channel programs that control the transmission of data. Each buffer pool handles storage for a different VTAM service; therefore, the needs of your network will determine which buffer pools are used the most. After you determine which buffer pools are the most important to your system, you may want to change the sizes of some of the buffer pools.

When you set a buffer pool size, you are reserving storage for use only by that buffer pool. If you specify buffer pools that are larger than necessary, you are setting aside storage that is not being used. In effect, the storage is being wasted when it could be used in other areas.

When you set a buffer pool size that is too small, VTAM will reach the buffer pool limit and will then dynamically allocate more space in the buffer pool. When the current need is satisfied, VTAM will then dynamically deallocate space in the buffer pool. Specifying small buffer pools conserves storage, but can cause frequent CPU use for expansion and contraction. For information on types of buffer pools and buffer pool allocation, see “Buffer Pools” on page 583.

NCP Slowdown

NCP slowdown is caused when NCP runs out of buffers for transferring data to VTAM. For information on what you can do to alleviate the slowdown condition, see “Controlling NCP Slowdown” on page 594.

Coattailing

Coattailing occurs when more than one piece of data is being transferred in or out of the host with a single channel input/output (I/O) operation. For example, without coattailing, each message read into VTAM generates a READ channel program. A READ channel program uses CPU cycles. If you can read more than one message into VTAM and generate only one read channel program, then coattailing is taking place and you are saving CPU cycles. Coattailing can provide greater throughput between a host and an SNA controller because fewer interrupts and instructions are processed for each message. For further information, see “Maximizing Coattailing” on page 595.

Tuning Tools

This section describes the tools that you can use to gather data to tune your network.

Monitoring Common Storage Areas

If availability of common service area (CSA), GCS common segment, or system GETVIS area (SGA) is a problem, you should monitor the VTAM buffer pools for unused storage. You can do this with the DISPLAY BFRUSE operator command or with the VTAM TYPE=SMS,ID=VTAMBUF trace. The output from this trace goes to the following:

```

MVS Generalized trace facility (GTF)
VM  TRSOURCE
VSE  SDAID.
```

The following example DISPLAY BFRUSE operator command report can be used to determine how much CSA is not being used.

```

D NET,BFRUSE
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = BUFFER POOL DATA
IST920I I000      BUFF SIZE  439      EXP INCREMENT   8
IST921I          TIMES EXP   0      EXP/CONT THRESH 6 / *N/A*
IST922I          CURR TOTAL  400      CURR AVAILABLE  363
IST923I          MAX TOTAL  400      MAX USED        277
IST924I -----
IST920I BS00      BUFF SIZE  146      EXP INCREMENT  25
IST921I          TIMES EXP   7      EXP/CONT THRESH 60 / 110
IST922I          CURR TOTAL  725      CURR AVAILABLE  97
IST923I          MAX TOTAL  775      MAX USED        708
IST924I -----
IST920I LP00      BUFF SIZE  2032     EXP INCREMENT   6
IST921I          TIMES EXP   0      EXP/CONT THRESH 1 / *N/A*
IST922I          CURR TOTAL   70      CURR AVAILABLE  66
IST923I          MAX TOTAL   70      MAX USED        24
IST924I -----
IST920I XD00      BUFF SIZE   673     EXP INCREMENT   5
```

```

IST921I      TIMES EXP  0      EXP/CONT THRESH 5  / *N/A*
IST922I      CURR TOTAL 6      CURR AVAILABLE  6
IST923I      MAX TOTAL  6      MAX USED        0
IST924I -----
IST920I LF00  BUFF SIZE 112    EXP INCREMENT  32
IST921I      TIMES EXP  0      EXP/CONT THRESH 1  / *N/A*
IST922I      CURR TOTAL 25    CURR AVAILABLE 13
IST923I      MAX TOTAL 25    MAX USED        12
IST924I -----
IST920I CRPL  BUFF SIZE 144    EXP INCREMENT 00025
IST921I      TIMES EXP  6      EXP/CONT THRESH 29 / 79
IST922I      CURR TOTAL 375   CURR AVAILABLE 154
IST923I      MAX TOTAL 450   MAX USED        418
IST924I -----
IST920I SF00  BUFF SIZE 104    EXP INCREMENT  34
IST921I      TIMES EXP  0      EXP/CONT THRESH 1  / *N/A*
IST922I      CURR TOTAL 51    CURR AVAILABLE 39
IST923I      MAX TOTAL 51    MAX USED        12
IST924I -----
IST920I SP00  BUFF SIZE 112    EXP INCREMENT  32
IST921I      TIMES EXP  0      EXP/CONT THRESH 1  / *N/A*
IST922I      CURR TOTAL 10    CURR AVAILABLE 10
IST923I      MAX TOTAL 10    MAX USED         2
IST924I -----
IST920I AP00  BUFF SIZE  56    EXP INCREMENT  56
IST921I      TIMES EXP  0      EXP/CONT THRESH 3  / *N/A*
IST922I      CURR TOTAL 16    CURR AVAILABLE 16
IST923I      MAX TOTAL 16    MAX USED         0
IST924I -----
IST449I CSALIMIT = NOLIMIT, CURRENT = 1623K, MAXIMUM = 2823K
IST790I MAXIMUM CSA USED = 2823K
IST449I CSA24 LIMIT = NOLIMIT, CURRENT = 16K, MAXIMUM = 36K
IST790I MAXIMUM CSA24 USED = 36K
IST595I IRNLIMIT = NOLIMIT, CURRENT = 0K, MAXIMUM = 0K
IST981I VTAM PRIVATE: CURRENT = 3294K, MAXIMUM USED = 3974K
IST314I END

```

The general formula for determining the amount of CSA not being used for a given buffer pool follows:

$$\text{NOTUSED} = (\text{MAXTOTAL} - \text{MAXUSED}) \times \text{BUFFSIZE}$$

Applying this formula to the I/O buffer pool shown in the sample DISPLAY BFRUSE command report, the amount of CSA not being used follows:

$$\text{IOBUF unused} = (400 - 277) \times 439 = 53,997 \text{ bytes of fixed CSA}$$

The unused CSA for the other pools shown follows:

```

BSBUF unused = 9782 bytes of fixed CSA
LPBUF unused = 93472 bytes of pageable CSA
XDBUF unused = 4038 bytes of fixed CSA
LFBUF unused = 1456 bytes of fixed CSA
CRPL  unused = 4608 bytes of pageable CSA
SFBUF unused = 4056 bytes of fixed CSA
SPBUF unused = 896 bytes of pageable CSA
APBUF unused = 896 bytes of fixed CSA

```

Therefore, the conclusion is as follows:

TOTAL CSA unused = 173,201 bytes
 TOTAL fixed = 74,225 bytes

As this example illustrates, this procedure can help you to identify where CSA is being not used so that you can adjust your usage.

Analyzing Slowdown

If you encounter a slowdown condition in your NCP, there are not enough buffers in the NCP to keep up with the amount of data being transferred. An NCP slowdown condition causes a message to be issued to the operator. However, you can also examine the SLODN tuning statistic to analyze the frequency that an NCP reaches a CWALL threshold condition, which can be encountered after the NCP has entered slowdown.

Slowdown in a channel-attached SNA cluster controller means that the controller (for example, the 3174) is out of buffers. The SLODN tuning statistic identifies when an SNA controller enters this condition.

Slowdown for a multipath channel connection indicates the number of times that read subchannel has gone into slowdown. Slowdown occurs when there is a shortage of IOBUFs when deblocking a multipath channel (MPC) read buffer.

Gathering Tuning Statistics

By using the TNSTAT start option or the MODIFY TNSTAT command, you can collect data that will help you set the proper values on resource definition operands that control VTAM I/O operations in your system.

You can use VTAM tuning statistics to gather information on communication between VTAM and any of the following channel-attached nodes:

- Communication controller
- Adjacent host
- SNA cluster controller.

Statistics are always recorded in the appropriate tuning statistics file:

- MVS** A system management facility (SMF) data set
- VM** FILE TUNSTATS A
- VSE** A trace file (an external file to which SYS001 has been assigned).

MVS You must include the system management facility (SMF) in the system during system generation.

VM Notes:

1. You must create an EXEC or application to process the output of tuning statistics. A sample EXEC is provided in SAMPLIB (IBM-supplied library).
2. An alternative file can be used by defining it to the TUNSTATS *ddname* with the FILEDEF command before VTAM is started. If you specify the DISP MOD option on the FILEDEF command, the statistics are appended to the end of the file, if the file already exists. If you do not specify DISP MOD, the existing file is overwritten, and only new statistics appear in the file.

For your convenience, a FILEDEF command has been included in the sample VMVTAM startup EXEC. It is commented out and can be uncommented and used.

3. You can print this file using the TUNING EXEC. This exec prompts you for the name of the file containing the tuning statistics. The default is FILE TUNSTATS. A sample of the TUNING EXEC is available on VTAM's 191 disk, which by default is MAINT's 298 disk.

You can also print this file in hexadecimal format by using the following command, where *file* is the default name FILE or the user-designated alternative file name:

```
PRINT file TUNSTATS A (HEX)
```

No tuning statistics are provided for LANs connected through XCA or ICA lines.

Tuning Input/Output (I/O) Operations for SNA Controllers

VTAM performs I/O operations to channel-attached communication controllers and cluster controllers.

You can use VTAM tuning statistics to adjust operands in NCP and VTAM definition statements to make your system run more efficiently.

VTAM uses channel programs to send data to SNA controllers. The amount of data that VTAM can read in one operation depends on the number of buffers used by a read channel program and on the size of each buffer.

Each tuning statistics record contains information about the state of data-transfer operations between VTAM and one channel-attached SNA controller (communication or cluster controller). Each record contains statistics that cover the time period since the last tuning statistics record was written for that controller or channel-to-channel connection.

Online Tuning Statistics Report for SNA Controllers: Following is an example of a tuning statistics report for an SNA controller that can appear at the VTAM operator's console:

```
IST440I    TIME = 07431380    DATE = 87190    ID = 0D0-L
IST441I    DLRMAX = 1        CHWR = 178        CHRD = 135
IST442I    ATTN = 31        RDATN = 0        IPIU = 196
IST443I    OPIU = 180        RDBUF = 196        SLODN = 0
IST314I    END
```

For a description of these fields, see message IST440I in Chapter 6, "IST Messages for VTAM Network Operators" in *VTAM Messages and Codes*.

Tuning Statistics File Report for SNA Controllers: If statistics are produced, they are always written to the appropriate tuning statistics file in the format shown in Table 32 and Table 33 on page 571.

Table 32 (Page 1 of 2). **MVS** **VM** Record Format for SNA Controller

Offset	Length	Format	Description
0	2	Binary	Record length
2	3	Binary	Reserved

Table 32 (Page 2 of 2). **MVS** **VM** Record Format for SNA Controller

Offset	Length	Format	Description
5	1	Binary	Record type
6	4	Binary	Time record moved to buffer
10	4	packed	Date: 00YYDDDF
14	4	EBCDIC	System identification
18	8	EBCDIC	Controller name
26	4	Binary	Dump load restart requests (DLRMAX)
30	4	Binary	Write channel program count (CHWR)
34	4	Binary	Read channel program count (CHRD)
38	4	Binary	Total attention interrupts received (ATTN)
42	4	Binary	Attentions on ends of READs (RDATN)
46	4	Binary	Number of PIUs inbound (IPIU)
50	4	Binary	Number of PIUs outbound (OPIU)
54	4	Binary	Total read buffers used (RDBUF)
58	4	Binary	Number of slowdowns (SLODN)

Table 33. **VSE** Record Format for SNA Controller

Offset	Length	Format	Description
0	2	Binary	Record length
2	6	character	Reserved
8	8	character	Time and date from STCK instruction
16	8	character	Controller name (ID)
24	8	character	Reserved
32	4	Binary	Dump load restart requests (DLRMAX)
36	4	Binary	Write channel program count (CHWR)
40	4	Binary	Read channel program count (CHRD)
44	4	Binary	Total attention interrupts received (ATTN)
48	4	Binary	Attentions on ends of read (RDATN)
52	4	Binary	Number of PIUs inbound (IPIU)
56	4	Binary	Number of PIUs outbound (OPIU)
60	4	Binary	Total read buffers used (RDBUF)
64	4	Binary	Number of slowdowns (SLODN)

Tuning I/O Operations for Channel-to-Channel Connections

VTAM performs I/O operations to other VTAMs across a channel-to-channel link. By analyzing statistics provided by VTAM, you can select I/O buffer sizes, data transfer delay, and other options that can improve performance of these I/O operations.

VTAM uses channel programs to send data to other hosts. The amount of data that VTAM can read in one operation depends on the number of buffers used by a read channel program and on the size of each buffer.

Each tuning statistics record contains information about the state of data-transfer operations between two VTAMs (using channel-to-channel adapters). Each record contains statistics that cover the time period since the last tuning statistics record was written for that controller or channel-to-channel connection.

Online Tuning Statistics for Channel-to-Channel Adapters: Following is an example of a tuning statistics report for a channel-to-channel attachment:

```
IST577I    TIME = 10214650    DATE = 94010    ID = LINEBC0
IST578I    CHNRM = 20        CHMAX = 0        RDBUF = 2091
IST579I    ATTN = 17        TIMERS = 0        QDPH = 14
IST580I    BUFCAP = 0        PRI = 6        SLODN = 0
IST581I    IPIU = 19        OPIU = 20        DLRMAX = 3
IST1022I   WRBUF = 1192
IST314I    END
```

For a description of these fields, see message IST577I in Chapter 6, "IST Messages for VTAM Network Operators" in *VTAM Messages and Codes*.

Tuning Statistics File Report for Channel-to-Channel Adapters: Table 34 and Table 35 on page 573 show the format of the record written to the appropriate tuning statistics file.

Table 34 (Page 1 of 2). **MVS** **VM** Record Format for Channel-to-Channel Adapters

Offset	Length	Format	Description
0	2	Binary	Record length
2	3	Binary	Reserved
5	1	Binary	Record type
6	4	Binary	Time record moved to buffer
10	4	packed	Date: 00YYDDDF
14	4	EBCDIC	System identification
18	8	EBCDIC	CTCA name (ID)
26	4	Binary	Dump load restart requests (DLRMAX)
30	4	Binary	Normal-sized channel program count (CHNRM)
34	4	Binary	Large-sized channel program count (CHMAX)
38	4	Binary	Total attention interrupts received (ATTN)
42	4	Binary	Number of write buffers used (WRBUF)
46	4	Binary	Number of PIUs inbound (IPIU)
50	4	Binary	Number of PIUs outbound (OPIU)
54	4	Binary	Total input bytes used (RDBUF)
58	4	Binary	Number of slowdowns (SLODN)
62	1	Binary	CTCA extension length (including this field)
63	1	Binary	CTCA attachment type
64	2	EBCDIC	CTCA version = X'F0F1'
66	4	Binary	Channel program starts due to timer trigger (TIMERS)

Table 34 (Page 2 of 2). **MVS** **VM** Record Format for Channel-to-Channel Adapters

Offset	Length	Format	Description
70	4	Binary	Channel program starts due to queue depth limit trigger (QDPH)
74	4	Binary	Channel program starts due to destination capacity limit trigger (BUFCAP)
78	4	Binary	Channel program starts due to high priority request trigger (PRI)

Table 35. **VSE** Record Format for Channel-to-Channel Adapters

Offset	Length	Format	Description
0	2	Binary	Record length
2	6	character	Reserved
8	8	character	Time and date from STCK instruction
16	8	character	CTCA name (CTCA)
24	8	character	Reserved
32	4	Binary	Dump load restart requests (DLRMAX)
36	4	Binary	Normal-sized channel program count (CHNRM)
40	4	Binary	Large-sized channel program count (CHMAX)
44	4	Binary	Total attention interrupts received (ATTN)
48	4	Binary	Number of write buffers used (WRBUF)
52	4	Binary	Number of PIUs inbound (IPIU)
56	4	Binary	Number of PIUs outbound (OPIU)
60	4	Binary	Total read buffers used (RDBUF)
64	4	Binary	Number of slowdowns (SLODN)
68	1	Binary	CTCA extension length (including this field)
69	1	Binary	CTCA attachment type
70	2	character	CTCA version = X'F0F1'
72	4	Binary	Channel program starts due to timer trigger (TIMERS)
76	4	Binary	Channel program starts due to queue depth limit trigger (QDPH)
80	4	Binary	Channel program starts due to destination capacity limit trigger (BUFCAP)
84	4	Binary	Channel program starts due to high priority request trigger (PRI)

Tuning I/O Operations for Multipath Channel Connections

With multipath channel attachment, VTAM performs I/O operations to other VTAMs across multiple single-direction channel links. By analyzing statistics provided by VTAM, you can select I/O buffer sizes, data transfer delay, and other options that can improve performance of these I/O operations.

VTAM uses channel programs to send data to other hosts. The amount of data that VTAM can read in one operation depends on the size of the read buffer.

Each tuning statistics record contains information about the state of data-transfer operations between two VTAMs (using multipath channel connections). Each record contains statistics that cover the time period since the last tuning statistics record was written for that multipath channel connection.

Multipath channel uses a special data block called a sweep that is exchanged with the adjacent host to verify that data has not been lost. A host initiating a sweep request holds all outbound multipath channel transmissions until it receives a sweep reply from the adjacent host.

A sweep is initiated when either of the following occurs:

- A timer expires in the host with the higher subarea number
- Receive queue depth in either host is excessive.

The host initiating the sweep sends the sequence number of the last output transmit block. The adjacent host compares this number with its last input transmit block sequence number. The adjacent host then sends a response to the initiating host that includes the adjacent host's last output transmit block sequence number. The initiating host makes the same comparison. If the numbers do not match, or the sweep does not complete within a time limit, the multipath channel group will be inactivated. Otherwise, normal flow continues. The tuning statistics contain a count of how many sweeps are initiated by an expired timer and how many are initiated by excessive receive queue depth.

Online Tuning Statistics for Multipath Channel Adapters: Following is an example of a tuning statistics report for a multipath channel attachment:

```
IST1230I TIME      = time      DATE      = date      ID = id
IST1231I IPIU      = ipiu      OPIU      = opiu
IST1232I TSWEPT    = tsweep    QSWEEP    = qsweep
IST1233I DEV       = dev       DIR       = dir
IST1234I BSIZE     = bsize     MAXBYTES  = maxbytes
IST1235I SIO       = sio       SLOWDOWN  = slowdown
IST1236I BYTECNT0  = bytecnt0  BYTECNT   = bytecnt
IST924I -----
IST1233I DEV       = dev       DIR       = dir
IST1234I BSIZE     = bsize     MAXBYTES  = maxbytes
IST1235I SIO       = sio       SLOWDOWN  = slowdown
IST1236I BYTECNT0  = bytecnt0  BYTECNT   = bytecnt
:
IST314I  END
```

For a description of these fields, see message IST1230I in Chapter 6, "IST Messages for VTAM Network Operators" in *VTAM Messages and Codes*.

Tuning Statistics File Report for Multipath Channel Connections: Table 36 on page 575 shows the format of the record written to the appropriate tuning statistics file. For each MPC group, a minimum of 3 records are written. The first record contains statistics for the entire MPC group. The subsequent records contain statistics for each write subchannel and each read subchannel.

Table 36. Record Format for Multipath Channel Connections

Offset	Length	Format	Description
0	2	Binary	Record length
2	3	Binary	Reserved
5	1	Binary	Record type
6	4	Binary	Time record moved to buffer
10	4	packed	Date: 00YYDDDF
14	4	EBCDIC	System identification
18	8	EBCDIC	MPC line name
26	4	Binary	Dump load restart requests (DLRMAX)
30	4	Binary	Number of inbound PIUs (IPIU) ¹
34	4	Binary	Number of outbound PIUs (OPIU) ¹
38	4	Binary	Number of timer sweeps (TSWEEP) ¹
42	4	Binary	Number of queue sweeps (QSWEEP) ¹
46	4	Binary	Number of write records
46	4	EBCDIC	Device address (DEV) ²
50	4	Binary	Number of read records ¹
50	4	Binary	Transmit block size (BSIZE) ²
54	4	EBCDIC	Subchannel polarity (DIR) ²
58	4	Binary	Transmit byte count (BYTECNT) ²
62	1	Binary	CTCA extension length (including this field)
63	1	Binary	CTCA attachment type X'01' ¹ X'02' ²
64	2	EBCDIC	CTCA version = X'F0F2'
66	4	Binary	Overflow byte count (BYTECNT0) ²
70	4	Binary	Slowdown frequency (SLOWDOWN) ²
74	4	Binary	Number of SIO issued (SIO) ²
78	4	Binary	Max transmit size (MAXBYTES) ²

Note:

1. ¹ indicates fields used only in the first record for statistics on the group.
2. ² indicates fields used only in subsequent records for statistics on each write and each read subchannel.

Analyzing Tuning Statistics

A single set of VTAM tuning statistics can be enough to indicate how a network is operating. However, these statistics become more valuable as you compare sets of values over time to see trends or the effects caused by changing buffer pool specifications and operands. You should analyze tuning statistics before and after making any change that might affect system performance.

Use tuning statistics to analyze the effects of VTAM host processor usage.

1. Compare the number of attention interrupts with channel READs. If the number of attention interrupts is the same as the number of channel READs, each READ is the result of the NCP sending the host a stand-alone attention. This condition causes increased CPU utilization because the host processor and VTAM must handle each attention interrupt. Conversely, if the number of attention interrupts is small in relation to the number of channel READs, the impact on the host is much less. If the number of attentions is large in relation to the number of channel READs (CHRD), then this could be a normal condition. Some controllers (for example, those running NCP Version 5 Release 2 and subsequent releases) send attention polls to the primary logical unit during periods of no traffic. VTAM does not count the number of channel reads when no data is transferred.
2. Analyze your IOBUF size. If RDBUF is much larger than IPIU, the value specified for the I/O buffer size might be too small. However, you need to know the message characteristics of your network before you make decisions. Many systems receive the bulk of their inbound data in very short messages and respond with longer outbound messages. For details on choosing an appropriate value for the I/O buffer size, see “Guidelines for Setting UNITSZ” on page 603.
3. Analyze data transfer operations between VTAM and an SNA controller. To analyze inbound data transfer, multiply MAXBFRU by the number of channel READs (CHRD). MAXBFRU is specified on the GROUP, HOST, LINE, or PU definition statement. The product should be close to the value of RDBUF (the number of read buffers used). Otherwise, data is not being transferred for all the channel command words (CCWs) in the read channel program. Reducing the value of MAXBFRU should solve this problem. However, the value of MAXBFRU times UNITSZ must be at least as large as the largest PIU received. For more details on setting the value of MAXBFRU and UNITSZ, see “Guidelines for Setting UNITSZ” on page 603 and “Guidelines for Setting MAXBFRU” on page 600.
4. Examine the read attention (RDATN) information that is available in VTAM tuning statistics to understand the efficiency of the inbound data transfer operations. RDATN is the number of times that VTAM, after reading data, is requested by an attention to read more data. This happens when one of the following occurs:
 - The READ channel program CCW string is not long enough to contain all of the data sent.
 - Enough additional data comes into the communication controller during the read operation to cause it to request VTAM to do another read.

Although this coattailing of PIUs is preferable to stand-alone attention interrupts, a large RDATN value is not desirable. To increase the size of the CCW

string, increase the MAXBFRU value. For more details on setting the value of MAXBFRU, see “Guidelines for Setting MAXBFRU” on page 600.

5. Analyze the outbound data transfer operation between VTAM and an SNA controller by analyzing VTAM's blocking of outbound PIUs. To determine the average number of PIUs for each write operation, divide the outbound PIU (OPIU) count by the number of channel WRITES (CHWR). This average number of PIUs indicates the effectiveness of the VTAM blocking algorithm. Increasing DELAY can also increase inbound coattailing. For more information on setting the value of DELAY, see “Guidelines for Setting DELAY” on page 598.

Determining the Amount of Coattailing in Your System

You can analyze the amount of coattailing in your system by examining VTAM tuning statistics. The tuning statistics indicate the total number of READ (CHRD) and WRITE (CHWR) channel programs that are used for data transfer operations. This information can be used to tune your system to improve coattailing if necessary. Increasing coattailing in your system can impact response times because coattailing means there are data transfer operations delays. When examining tuning statistics, also consider that short periods of high-attention counts can simply indicate a light load on the network.

Migrating from User-Replaceable Constants

User-replaceable constants were formerly used to control various VTAM functions, such as:

- Buffer use storage management services (SMS) trace “snapshot” value
- Maximum RU size for sessions
- BSC 3270 timeout (inoperative) value
- Virtual-route-selection subtask interval analysis block
- Host intermediate routing node (IRN) slowdown storage
- NetView trace data buffers
- Directory size of the symbol resolution table for the network.

User-replaceable constants are now set only as start options. For additional information, refer to Appendix H, “Migrating from Use of VTAM Constant Values to V4R2 Start Options” in *VTAM Customization*. For more information on start options, see Chapter 4, “Start Options” in the *VTAM Resource Definition Reference*. Start options can be displayed with the DISPLAY VTAMOPTS command, and many of them can be changed with the MODIFY VTAMOPTS command. For more information on the DISPLAY VTAMOPTS and MODIFY VTAMOPTS commands, see “MODIFY VTAMOPTS Command” and “DISPLAY VTAMOPTS Command” in *VTAM Operation*.

Estimating the Number of Active Sessions

The EAS operand on the LU and APPL definition statements specifies the estimated number of active sessions between VTAM and the application program or VTAM and the logical unit.

EAS Operand for Application Programs

The EAS operand on the APPL definition statement indicates the amount of storage to allocate for a table which contains information on any sessions with the application program. The EAS value that you choose can affect performance in your system.

Specifying an EAS value that is too small can cause an increase in path length for VTAM processing of inbound data flows. The increase is due to VTAM scanning control blocks for some of the sessions. As more sessions become active, the directory entries are used more frequently, and the chains become longer. Longer chains require more scan time. If you specify a larger EAS value, the chains for particular directory entries may be smaller and may use less CPU time for scanning the chains.

For a non-TSO application program, you should specify an EAS value that equals the average number of sessions, but does not exceed 32767. If you have virtual storage constraints, you can specify a smaller EAS value.

MVS For a TSO application program, you should specify EAS=1.

EAS Operand for Independent Logical Units

For performance reasons, if you anticipate that a particular logical unit can have more than 256 sessions, increase the value of the EAS operand to the nearest multiple of 256. For example, if you anticipate 700 sessions for a particular logical unit, you should define EAS to be 768 (3 x 256). Coding a poor estimate for EAS does not affect your sessions, but it does affect performance.

You can define the EAS operand for an independent logical unit only.

Common Storage Areas

Common storage areas are used by VTAM and other resources in your network. The types of common storage areas are:

- Common Service Area (CSA) for MVS systems
- GCS common segment for VM systems
- System GETVIS Areas (SGA) for VSE systems.

Common Storage Areas for MVS and VM

VTAM maintains buffers and control blocks in the common service area (CSA) for MVS systems and GCS common segment for VM systems. For performance reasons, VTAM offers you some control over its use of common storage areas.

Common Service Area Limit

The CSALIMIT start option specifies the maximum amount of common service area (CSA) that is used by VTAM. It can be used to prevent VTAM from using CSA that is needed by the operating system. Most environments do not need to limit VTAM's use of CSA because their peak CSA usage is a very small increment over their requirements when all of their logical units are in session.

If the limit specified by CSALIMIT is reached, the results are unpredictable. If LPBUF cannot be expanded, VTAM might enter a deadlock condition. Therefore, if you specify the CSALIMIT start option, you should define LPBUF large enough so

that it does not have to expand. Other possible occurrences are that messages can be lost or a session initiation or termination can fail.

Common Service Area 24-Bit

The CSA24 start option specifies the maximum amount of 24-bit addressable common service area (CSA) that can be used by VTAM. The largest possible value for CSA24 is 16 megabytes.

The CSA24 start option limits only the amount of 24-bit addressable storage explicitly requested by VTAM. The CSA24 limitations do not apply when MVS returns 24-bit addressable storage in response to a VTAM storage request above the 16-megabyte line. MVS returns 24-bit addressable storage below the line because not enough CSA above the line is defined. CSA storage can be allocated from above or below the line depending on where the allocating procedure resides.

Common Storage Areas for VSE

When VTAM resides in a private partition, it allocates control blocks and storage areas in the SGA to initialize user applications, to schedule VTAM deferred processes, and to process application requests.

System GETVIS Area

You can use the SGALIMIT start option to specify the maximum amount of SGA that VTAM can use. You can also use the MODIFY SGALIMIT to change the SGA limit while VTAM is running.

If the limit you specified on the SGALIMIT start option is reached, VTAM will be unable to obtain further storage from the SGA and may not be able to complete the operation for which it was obtaining storage.

To avoid this situation, you should plan carefully to make sure the SGALIMIT is sufficient to meet VTAM's needs. Use the DISPLAY BFRUSE command to monitor your SGA needs.

System GETVIS Area 24-Bit

The SGA24 start option limits only the amount of 24-bit addressable storage explicitly requested by VTAM. The SGA24 limitations do not apply when VSE returns 24-bit addressable storage in response to a VTAM storage request above the 16-megabyte line. VSE returns 24-bit addressable storage below the line because not enough GSA above the line is defined. GSA storage can be allocated from above or below the line depending on where the allocating procedure resides.

DISPLAY STORUSE Pools

Table 37 on page 580 provides a list of storage pools that are displayed using the DISPLAY STORUSE command. Included for each pool is a short description of the pool's function and characteristics. These pools are not customer-defined, unlike the buffer pools defined via the VTAM start options (for example, IOBUF). VTAM allocates and deallocates storage from these pools as needed.

If VTAM is in a storage shortage situation, Table 37 on page 580 and the output from the DISPLAY STORUSE command can be used to determine where excess storage is being used, enabling you to take appropriate action to remedy the shortage.

Table 37 (Page 1 of 4). DISPLAY STORUSE Pools

POOLNAME	Pool # (Hex)	Storage Location	Description
ACDEB	07	SYSTEM	A pool element is allocated for every active application.
ACPCB	50	USER	An element is allocated for every adjacent control point with which this node has an active CP-CP session.
ADJCP	48	USER	Each pool element defines a single adjacent control point (ADJCP).
ADJNODE	57	USER	Elements are allocated for each CP-CP session partner to track topology flow status with an adjacent node.
AMU	23	SYSTEM	Elements are short-lived signals used for intra-product communication.
ANDCB	54	USER	One element is allocated whenever a node activates a CP-CP session pair with an adjacent node.
ATGB	49	USER	An element represents a single T2.1 connection or VR-TG to an adjacent CP.
AUTOLOGN	2A	USER	Elements are used to keep track of autologon relationships.
BFRTRACE	32	SYSTEM	Elements are used to hold small buffer trace records.
BFRTRFUL	66	SYSTEM	Elements are used to hold large buffer trace records.
BNINFO	6A	USER	Elements are used to perform Border Node search processing.
BSBEXT	6C	USER	Elements are allocated for each session using SNA/IP support.
CAB	30	USER	One element is allocated for each VCNS connection resulting from a LOGON to a VCNS line (LANs) or from an X.25 OPEN command.
CACHE	58	USER	Elements are used for caching PCIDs during direct search list processing.
CANT	2F	USER	One element is allocated for every 64 VCNS X.25 connections for the same LOGON to a VCNS line.
CDAJSCP	1C	USER	Elements are used to define adjacent SSCP entries.
CDRSC	06	USER	Elements are used to define dynamic CDRSCs.
CNSFACUD	31	USER	An element is allocated every time a VCNS application asks VTAM to initiate or receive an X.25 call request that will contain facilities or call user data.
CORCB	5B	USER	Elements are used for APPN Locate request correlation.
COS	4D	USER	Elements are used for APPN COS definitions and mode table mappings.
CPRUPE	53	USER	Elements are request/response unit processing elements used when processing APPN-related requests.
CPWACSA	2B	SYSTEM	An element is allocated when a USS command is entered from the network operator's console or from a user terminal. One element from the pool is allocated when an application resource definition specifying SSCPFM=USSNOP is being processed.
CPWAPVT	19	USER	One element is allocated when a USS command is entered from the network operator's console or from a user terminal. One element from the pool is allocated when an application resource definition specifying SSCPFM=USSNOP is being processed.
DCX	46	SYSTEM	Pool elements are used to maintain data compression information.
DDEL	72	USER	Elements are used to delay the disconnection of a PU that is defined with the DISCNT=DELAY parameter.
DECB	51	USER	An element is allocated for each resource in the APPN directory database.
DISKIO	55	USER	One element is allocated per component performing database hardening. The storage for this pool is allocated below the 16M line.
DMTSQ	33	USER	One element is allocated every time a message contained in the message flooding table is issued. The allocated element is freed when the message's suppression time expires.
DSERVER	56	USER	Elements are control blocks and short-term signals related to directory services and interchange nodes.
DSSIB	1F	USER	An element is allocated when a DSRLST is received and freed when the DSRLIST response is sent.
DYPATH	68	USER	Elements are path table entries for dynamic PUs.
EPTDVT	05	SYSTEM	Elements are used to contain DLC-specific information.
ERICPOOL	1A	USER	Elements are used while parsing request/response units. When the RU processing completes, the elements are freed.
ERTE	09	USER	Elements are used to define explicit routes resulting from PATH statements.
FMCB	34	SYSTEM	Elements are allocated once for each session associated with an application LU (for non-persistent LU sessions) at session BIND time.
FMCBEXT	0B	SYSTEM	Elements are allocated once for each session associated with an application LU at session BIND time.
FMH5	3D	SYSTEM	One element is allocated for each incoming LU 6.2 conversation request.

Table 37 (Page 2 of 4). DISPLAY STORUSE Pools

POOLNAME	Pool # (Hex)	Storage Location	Description
GRINS	6B	USER	Elements are used for to maintain associations between network resources and generic names.
GWNAJSCP	1D	USER	Elements are • Adjacent SSCP routing tables used to route CDINIT or DSRLST RUs. Elements are freed when the routing completes or fails. • Information to determine the gateway NCP to use during session setup.
HSICB	24	SYSTEM	Elements are allocated once for each APPC session associated with an application LU at session BIND time.
HSQH	08	SYSTEM	Elements are allocated once for the first of every 107 sessions set up across a VR, at session BIND time.
IAP	6F	USER	Elements are used to monitor data traffic between two internet address pairs. There is one IAP element for every pair of internet addresses which have a TCP/IP connection established between them. The pool element is retained as long as there is at least one connection between the internet address pair.
IOBLOCK	0F	SYSTEM	Elements in the pool are DLC-related control blocks (e.g. for a channel-attached NCP). The storage for this pool is fixed and is not paged out of memory by the operating system.
IOSIB	1E	USER	Elements are used to process Init_Other (Cross-domain) requests.
IPWKE	71	USER	Elements are signals used for intra-product communication. The number of these short-lived elements depends on the number of sessions into the TCP/IP network.
ISTENDEL	5A	USER	One element is allocated by a network node for each adjacent served end node or non-native network node.
ISTSITCB	59	USER	One element is needed at endpoint and network node server roles for each APPN Search procedure.
ISTTRCEL	11	USER	Elements are used for the problem determination (PD) trace function of the CNM interface.
KEYTOKEN	4B	USER	Elements map active cryptographic session key tokens.
LCB	5C	USER	An element is allocated to process Locate Search requests.
LIA	70	USER	One element is allocated per local address being used by the host. The element exists as long as there is at least one TCP/IP connection on the local address.
LMTABLE	25	SYSTEM	Three types of elements come from this pool: 1. One element is allocated for every pair of LUs that have negotiated APPC session limits. The element is allocated when a CNOS with a partner LU is initiated, and freed when the application closes its ACB. 2. An element is used for every application that opens its ACB with APPC=YES. 3. Elements represents current session limits between two LU 6.2s on a particular mode. An element is allocated for every logmode that has been negotiated between two partner LUs. The elements are freed for deletion, or freed when the application closes its ACB.
MIWKE	6E	USER	Elements are allocated for each active PU in a TCP/IP major node.
NDREC	4E	USER	Elements are used for APPN topology node information.
NIDCB	52	USER	An element is allocated for each network identifier known to the APPN Directory Database.
NQDAT	04	USER	One element allocated for each network-qualified SRTE.
NSRUL	41	SYSTEM	Elements are used to process LU 6.2 session activation. NSRUL is used for larger-sized requests.
NSRUS	40	SYSTEM	Elements are used to process LU 6.2 session activation. NSRUS is used for smaller-sized requests.
NSSCB	39	SYSTEM	Elements are used to process LU 6.2 session activation.
OBTSCB	3E	USER	Elements are allocated when a band datagram is created or read from the TCP/IP network and freed when the datagram is written to the TCP/IP network. Elements are also allocated for expedited flows or disconnect flows for sessions across TCP/IP using SNA/IP support.
OSCB	5D	USER	Elements are used to track outstanding Locate search requests.
PAQ	17	USER	Elements are used to track PLU network addresses for a given LU.
PCDCA	69	USER	Elements are used for border node PCID caching.
PGIOBLK	2C	USER	Elements are used when communicating with the 3172.
PLOCB	60	USER	Elements are used to process Locate search requests and replies.
PLUSC	38	SYSTEM	Elements are used during persistent LU session CLOSE processing.
PLUSDATA	47	SYSTEM	Elements are allocated once for each persistent LU session associated with an application LU, at session BIND time.
PLUSFMCB	35	SYSTEM	Elements are allocated once for each persistent LU session associated with an application LU, at session BIND time.

Table 37 (Page 3 of 4). DISPLAY STORUSE Pools

POOLNAME	Pool # (Hex)	Storage Location	Description
POAPRIV	65	USER	One element is allocated for each message destined for a Program Operator Application (POA). If the message requires a reply, a second POAPRIV element is allocated.
POWECOMM	15	SYSTEM	One element is allocated for every message issued when VTAM is running under a user task.
POWEPRIV	14	USER	One element is allocated for every message issued when VTAM is running under the VTAM task.
POWMCOMM	64	SYSTEM	One element is allocated for every single-line message and for every message group when VTAM is running under a user task.
POWMPRIV	63	USER	One element is allocated for every single-line message and for every message group when VTAM is running under the VTAM task.
PRDLE	2D	SYSTEM	Elements represent random data being used for establishing LU 6.2 sessions with session-level security.
PRIDBLK	28	USER	Elements map Procedure-Relation Identifier Blocks (PRIDs)
PRIDQAB	29	USER	Elements are used to maintain Procedure-Relation Identifier Blocks (PRIDs).
PULURDTE	16	USER	Elements are used to define dynamic PUs and LUs.
PXBFIXED	36	SYSTEM	Elements are used to expand fixed buffer pools. The storage for this pool is fixed and is not paged out of memory by the operating system.
PXBPAGED	37	USER	Elements are used to expand pageable buffer pools.
RAB	27	SYSTEM	One element is allocated for each APPC conversation.
RAQ	18	USER	Elements are used to queue requests when a usable network address is not available and an RNAA must be sent.
RIBRANT	2E	USER	One element is allocated for each LOGON of a VCNS application to a VCNS line, plus one extra element for every 16 LOGONS.
RUCON	42	SYSTEM	One element is allocated each time a DISPLAY ROUTE,TEST=YES command is issued. After 168 elements have been allocated, VTAM will delete any elements that have not been used for more than 30 minutes.
RUPECOMM	01	SYSTEM	Elements are allocated from this pool to process Request/Response Units (RUs) when execution is taking place in the VTAM address space.
RUPEPRIV	00	USER	Elements are allocated from this pool to process Request/Response Units (RUs) when execution is taking place in a non-VTAM address space.
SAB	26	SYSTEM	One element is allocated for each APPC session.
SCCB	5E	USER	Elements are used to perform search concentration.
SIB	02	USER	One element is allocated for each LU-LU session.
SIBEXT	0C	USER	One or two elements are allocated for each cross-network session. The elements are freed when the session ends.
SIBIX	1B	USER	One element is allocated during session initiation. The element is freed when the session becomes active.
SLD	3F	SYSTEM	Elements are allocated to process APPCCMD CONTROL=OPRCNTL,QUALIFY=DISPLAY macro instructions, including those issued from the operator console.
SLENT	67	USER	A pool element is allocated when a CPSVCMG session is activated between an End Node and serving Network Node, or for any CP SNASVCMG sessions that VTAM's Management Services Transport activates.
SOCCBEXT	73	USER	One element is allocated per session that goes across a TCP/IP network. (Contains fields which do not need to be in SYSTEM storage)
SOCKET	6D	SYSTEM	One element is allocated per session that goes across a TCP/IP network.
SRTE	10	USER	Elements are entries in the symbol resolution table.
SSCPFMCB	03	USER	One element is allocated for each SSCP-PU or SSCP-LU session.
STB	43	USER	Elements store information concerning a T2.1 adjacent link station that an independent LU is using for session connectivity.
TGP	4A	USER	One element is created for each TG profile (TGP) entry.
TGREC	4F	USER	Elements are used for APPN topology TG information.
TREEBLD	61	USER	Elements are used for APPN routing tree construction and maintenance.
TRSINFO	4C	USER	Elements are used for topology broadcast lists (for use during topology database update broadcasting) and endpoint TG vector information (during route calculation).
UECB	0E	SYSTEM	One element is allocated each time a user exit is to be scheduled.
UTILAREA	5F	USER	Elements are used to perform Locate search processing.
UTILCSAL	22	SYSTEM	Generic utility pool for large blocks of storage that must be in CSA.

Table 37 (Page 4 of 4). DISPLAY STORUSE Pools

POOLNAME	Pool # (Hex)	Storage Location	Description
UTILCSAS	21	SYSTEM	Generic utility pool for small blocks of storage that must be in CSA.
UTILFIXD	62	SYSTEM	Elements are used to map Route Selection Control Vectors (RSCVs) received during session initiation. The storage for this pool is fixed and is not paged out of memory by the operating system.
UTILPVTL	20	USER	Generic utility pool for large blocks of storage that must be in VTAM private storage.
UTILPVTS	12	USER	Generic utility pool for small blocks of storage that must be in VTAM private storage.
UVRPL	45	USER	One element is allocated each time a user exit is to be scheduled.
VRPL	13	SYSTEM	Elements are VTAM's copy of an application's Request Parameter List (RPL). Elements are also used for BINDs and other RUs received from the network.
WAR	44	USER	Elements represent autologon session originators (OLU-SLU) waiting for the availability of a required PU resource.
WREEID	0A	USER	Elements are used to suspend and resume VTAM processes.

Note: Unless otherwise specified:

- All pool storage is pageable and can be paged out of memory by the operating system.
- All pool storage can be located above or below the 16M line.

Buffer Pools

VTAM uses buffer pools to control the handling of data. It dynamically allocates and deallocates space in these buffer pools for the VTAM control blocks, I/O buffers, and channel programs that control the transmission of data. Specifying large buffer pools can waste storage, but does not require frequent CPU use for expansion. Specifying small buffer pools conserves storage, but requires frequent CPU use for expansion and contraction. With proper tuning, you can achieve a balance between storage use and performance that is suitable for your environment.

Types of Buffer Pools

Table 38 lists the types of buffer pools.

MVS, VM All buffer pools are allocated in extended CSA or GCS subpool 231 with KEY=6.

VSE All buffer pools are allocated in SGA except XDBUF, which is allocated in the partition.

Table 38 (Page 1 of 3). VTAM Buffer Pools

Buffer Pool	System	Use	Default Storage Type	Recommendations and Requirements
APBUF	MVS	Used to provide fixed common storage if VTAM fails to obtain other storage from the operating system. For buffers not related to I/O.	Fixed	IBM-supplied values are appropriate for most systems.
BSBUF	All	Used to maintain session information for peripheral nodes for which VTAM performs boundary function. One buffer is required for each boundary LU session (SSCP-PU, SSCP-LU, LU-LU).	MVS, VSE - Fixed, VM - Pageable	Set base number to the average number of concurrent boundary LU sessions.

Choosing Buffer Pool Specifications

Table 38 (Page 2 of 3). VTAM Buffer Pools

Buffer Pool	System	Use	Default Storage Type	Recommendations and Requirements
CRPLBUF	All	RPL-copy pool. One buffer is required for each VTAM application program request until the operation is complete.	Pageable	Monitor output from the DISPLAY BFRUSE command to ensure that the number of buffers specified is adequate for your system.
IOBUF	All	Used for input/output data. Every PIU that enters or leaves VTAM resides in an I/O buffer. MVS, VM This pool is 31-bit addressable. VSE This pool is 24-bit addressable.	Fixed	Monitor output from the DISPLAY BFRUSE command to ensure that the number of buffers specified is adequate for your system. Ensure that the largest MAXBFRU value is less than <i>xpanpt</i> minus <i>slowpt</i> . Also ensure that <i>bufsize</i> is greater than or equal to the UNITSZ operand on the HOST definition statement in the NCP.
LFBUF	All	One buffer is required for each active application program with an EAS value (on APPL definition statement) less than 30. If the EAS value is greater than 30, this information is contained in SFBUF. MVS - One buffer is required for each TSO user who is logged on.	MVS, VSE - Fixed, VM - Pageable	IBM-supplied values are appropriate for most systems.
LPBUF	All	Used for scheduling and audit trail (error recovery). One buffer is required for each active VTAM process.	Pageable	MVS - Set <i>baseno</i> to 9, and set <i>xpanno</i> in the range 4 – 6 to force 2-page expansion. VM and VSE - IBM-supplied values are appropriate for most systems.
SFBUF	All	Used to contain application program information and LU blocks. One buffer is required for each active application program with an EAS value (on APPL definition statement) greater than or equal to 30. VSE - One buffer is required for each active application program.	Fixed	IBM-supplied values are appropriate for most systems.
SPBUF	All	Used for large message (LMPEO) requests. One buffer is required per concurrent LMPEO send request.	Pageable	IBM-supplied values are appropriate for most systems.

Table 38 (Page 3 of 3). VTAM Buffer Pools

Buffer Pool	System	Use	Default Storage Type	Recommendations and Requirements
XDBUF	All	Used for physical I/O to VTAM peripheral nodes for connection activation (for example, XID exchange processing). MVS This pool is 31-bit addressable. VM, VSE This pool is 24-bit addressable.	Fixed	IBM-supplied values are appropriate for most systems. Because this buffer pool handles the concurrent activation of physical units, you should specify a small size and enable dynamic buffering. The buffer pool can then expand to accommodate the concurrent activation and can shrink after the resources are active.

Buffer Pool Allocation

You can use two types of buffer pool storage allocations, basic and dynamic.

Basic allocation The amount of space that you reserve for the buffer pool when VTAM is started.

Dynamic allocation The process by which VTAM temporarily increases the size of a buffer pool when there are heavy demands for space in that pool.

While basic allocation sets the base size of your buffer pools, dynamic expansion enables a buffer pool to be expanded temporarily during periods of heavy demand. Its use can greatly increase the efficiency with which VTAM uses storage, particularly for I/O buffers.

Without dynamic expansion of a pool, basic allocation parameters must be specified large enough to meet the greatest possible demands on the pool. With dynamic expansion, smaller basic allocation values can be specified, and peak demands on the pool can be met with dynamic expansion. Dynamic expansion is strongly recommended for most buffer pools because the peak demand can vary considerably from the normal demand.

Setting Buffer Pool Allocations

To set buffer pool space, use the buffer pool start options:

```
poolname=(baseno,bufsize,slowpt,F,xpanno,xpanpt,xpanlim)
```

For a description of these operands, see “Buffer Pool” in the *VTAM Resource Definition Reference*.

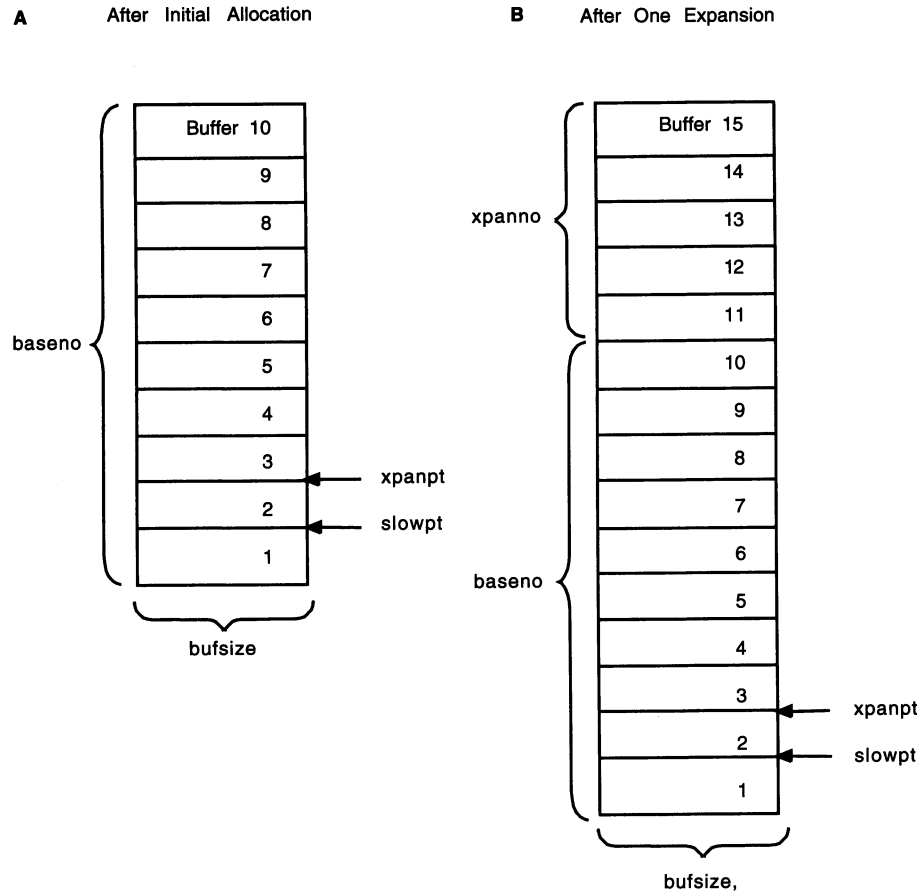
The first four operands of the start option (*baseno*, *bufsize*, *slowpt*, and *F*) establish the base size and the characteristics of the buffer pool. The last three operands (*xpanno*, *xpanpt*, and *xpanlim*) specify how dynamic expansion of the buffer pools is to be performed.

Set basic allocation for each buffer pool when VTAM is started. If you do not specify a start option parameter for a buffer pool, VTAM uses an IBM-supplied value for the missing parameter. The IBM-supplied values are not necessarily the most efficient values for your system, and they are not necessarily compatible with any dynamic allocation specifications you might make.

How Dynamic Expansion Operates

1. You enable dynamic expansion by coding the *xpanno* and *xpanpt* operands on the buffer pool start option.
2. The buffer pool size reaches the point specified by *xpanpt*.
3. VTAM acquires more buffers in blocks to accommodate the increased demand. The size of the blocks is specified by *xpanno*. VTAM acquires buffers as needed until the pool size reaches *xpanlim* (if specified).
4. Contraction is determined by *xpanno* and *xpanpt*. The *xpanno* value used is the *xpanno* value that was specified when VTAM was started, but rounded upward to the number of buffers that fills the nearest whole page of storage. When the number of available buffers is greater than or equal to $(2 \times xpanno) + xpanpt$, VTAM verifies that the buffers that it acquired in previous expansions of the pool are not in use. If they are not, VTAM releases these buffers in blocks (as when they are acquired). If any of the buffers in a block are in use, VTAM does not release that block of buffers.

Figure 148 on page 587 shows the structure of a pool after basic allocation **A** and after one dynamic expansion of the pool **B**.



- A** This example shows a buffer pool for which the start options were specified as `poolname=(10,bufsize,1,,5,2,xpanlim)`. After initial allocation, the pool contains 10 buffers (`baseno=10`), the length in bytes of each buffer is `bufsize`, the slowdown point (`slowpt`) is 1, the expansion size (`xpanno`) is 5 (assume that five buffers fill one page of storage), and the expansion point (`xpanpt`) is 2. The maximum allowed size of this buffer pool is determined by the value `xpanlim`.
- B** After one expansion, there are 15 buffers in the pool. Each of the five additional buffers has a length of `bufsize` and the same expansion point and slowdown point as before.

Figure 148. Buffer Pool after Initial Allocation and after One Expansion

Guidelines for Dynamic Expansion

The following guidelines help you determine appropriate buffer pool values for your system.

Note that if either `xpanno` or `xpanpt` is 0, no expansion occurs.

Set `baseno` for the I/O buffer to the steady state value plus `xpanpt`.

You can determine steady state by issuing the `DISPLAY BFRUSE` operator command. `CURR TOTAL` minus `CURR AVAIL` gives the number of buffers in use. Repeat this procedure several times, and take the average to determine the steady state value. Be sure to include in your calculation displays taken during times of peak network usage.

Any additional buffers needed can be obtained using dynamic expansion.

Define dynamic expansion so that pools are expanded one page at a time.

To do this, define *xpanno* to be the number of buffers that fit on one page. To calculate *xpanno* for I/O buffers, use Table 40 on page 590, and do the following:

1. Find the *bufsize* range in the table that includes the *bufsize* value that was specified for your system.
2. Round up the current *bufsize* to the largest even number that appears in that range. (Some channel-attached SNA devices require that the I/O *bufsize* be an even number of bytes. Therefore, you should define *bufsize* as an even number.)
3. For one-page expansion, specify an *xpanno* value that is equal to the number in the left column of this row (the number of I/O buffers for each page).

Tune the expansion increments (*xpanno*) and expansion points (*xpanpt*) to keep CPU overhead from dynamic buffering low.

- Expansion and contraction of buffer pools increase the amount of real storage that VTAM requires. If the buffer pools are continually expanding and contracting, you should change the values specified for *xpanno* and *xpanpt*. You can determine what values to specify by monitoring the output of the DISPLAY BFRUSE operator command or the SMS buffer trace and by modifying the values for *xpanno* and *xpanpt* accordingly.
- If the expansion point is set too small, VTAM might not be able to expand the buffer pool fast enough if there is a rapid demand for buffers. Setting the expansion point too small also causes the buffer pool to contract frequently.

Message IST561I might indicate that storage is temporarily unavailable due to this rapid demand. Adjust the *xpanpt* and *xpanno* values to eliminate this problem.

The expansion point for I/O buffers should be larger than the largest single request for buffers. Following are examples of the largest single request:

- MAXBFRU allocation
- NetView session monitor trace buffer size
- **MVS** JES/NJE TPBFSIZ.

- If excessive expansion and contraction are a problem (especially for LPBUF and CRPLBUF), define these buffers to expand two pages at a time.

The buffer pool can expand and contract too frequently if the expansion increment and expansion point are not large enough to handle normal fluctuation in the pool. The I/O buffer pool is prone to this type of thrashing from application programs that send very large messages. VTAM always rounds up an expansion increment to fill a multiple of a full page.

Initially allocate one page to SFBUF.

Any elements (where an element consists of a buffer, whose size is rounded to a multiple of 8 bytes, plus a 16-byte header) that are required over a page should be controlled by dynamic expansion.

Set CRPLBUF.

In environments where many sessions come up simultaneously because of either the VARY LOGON operator command or the LOGAPPL operand on a definition statement, you might need to pre-allocate enough CRPLBUF buffers to handle the influx of requests.

Set LFBUF.

LFBUF is used only for EAS values (on APPL definition statements) less than 30. Because this pool is not used often, a *baseno* value of 2 is recommended. **MVS** If you are using TSO, a buffer from the LFBUF buffer pool is used for each TSO user who is logged on. The pool should be expanded one page at a time.

Specify *xpanlim* for the IOBUF pool

xpanlim is useful in constraining the IO buffer pool from excessive growth particularly when a logical unit attempts to flood VTAM with requests (HOT I/O). Typically, the logical unit is malfunctioning, however the logical unit may be functioning correctly but may not have had proper session parameters specified. Specifying *xpanlim* allows VTAM to detect such sessions prior to complete CSA exhaustion and, if the HOTIOTRM Start Option is also specified, terminate such sessions without the system being impacted.

To set *xpanlim*, issue DISPLAY BFRUSE,BUFFER=IO find the peak usage, and set the limit somewhat above that peak. Because *xpanlim* is specified in units of 1024 (K) bytes, you will need to convert buffers to K bytes. Use Table 40 on page 590 to determine the number of IO buffers of a given size that fit on a 4K page. Use the equation $K = (\text{number of buffers} / \text{number of buffers per page}) * 4$.

Also, bear in mind that in order for the expansion limit to be meaningful, CSA must be available were the expansion limit to be reached and VTAM must not reach any CSA limit value that has been specified. In other words, ensure that you have adequate CSA available in the system and your CSA limits allow VTAM to actually allocate buffers up to your expansion limit.

Note: In Table 40 on page 590 where it lists a range of buffer sizes for a given buffers per page value, whenever possible, use the highest value in the range. Some devices have specific IO buffer size requirements that may prevent using the highest value.

VM VTAM can expand buffer pools from any machine in a GCS group that requires it. If you want VTAM to expand a locked buffer pool from any machine, then all machines that need to request expansion must have both the DIAG98 directory option specified and a virtual storage size large enough to contain the entire GCS shared system. If you use this option, VTAM does not use any more locked pages than it would otherwise. If VTAM cannot expand the locked buffer pool in a machine in the GCS group, then the expansion of the buffer pool is scheduled in the VTAM machine.

Table 39. Number of Buffers per Page

Buffer Pool	MVS	VM	VSE 4K Paging
APBUF	56	N/A	N/A
BSBUF	24	24	24
CRPLBUF	25	25	24
LFBUF	32	32	32
LPBUF	2	2	2
SFBUF	34	34	34
SPBUF	28	28	28
XDBUF	5	5	5

Table 40 (Page 1 of 2). I/O Buffer Size and Number of Buffers per Page

Number of Buffers per Page	Range of buffer sizes for MVS (with 4K paging)	Range of buffer sizes for MVS using data encryption facility (with 4K paging)	Range of buffer sizes for VM and VSE (with 4K paging)
1	4008–1962	4000–1955	4016–1970
2	1961–1274	1954–1267	1969–1282
3	1273–938	1266–931	1281–946
4	937–730	930–723	945–738
5	729–594	722–587	737–602
6	593–498	586–491	601–506
7	497–426	490–419	505–434
8	425–362	418–355	433–370
9	361–322	354–315	369–330
10	321–282	314–275	329–290
11	281–250	274–243	289–258
12	249–226	242–219	257–234
13	225–202	218–195	233–210
14	201–186	194–179	209–194
15	185–170	178–163	193–178
16	169–154	162–147	177–162
17	153–138	146–131	161–146
18	137–122	130–115	145–130
19	121–114	114–107	129–122
20	113–106	106–99	121–114
21	105–98	98–91	113–106
22	97–90	90–83	105–98
23	89–82	82–75	97–90
24	81–74	74–67	89–82

Table 40 (Page 2 of 2). I/O Buffer Size and Number of Buffers per Page

Number of Buffers per Page	Range of buffer sizes for MVS (with 4K paging)	Range of buffer sizes for MVS using data encryption facility (with 4K paging)	Range of buffer sizes for VM and VSE (with 4K paging)
25	73–66	66–64	81–74
26	65–64	–	73–66

Dynamic Expansion and Storage

As buffer pools expand, expansion uses virtual storage that can potentially impact an operating system that has virtual storage constraints. VTAM's use of virtual storage can be limited by the start options and the virtual storage allocated to VTAM's address space (MVS), virtual machine (VM), or partition (VSE).

All buffer pools are defined in virtual storage above the 16-megabyte address line in extended virtual storage. Therefore, the size of the buffer pools does not usually impact virtual storage constraints. **VSE** IOBUF and XDBUF buffer pools are defined below the 16-megabyte address line.

You can use the SONLIM start option to limit the amount of fixed I/O buffer storage available for session outage notification messages. This controls VTAM's use of fixed (real) storage. The default (60 percent) should be sufficient for most environments. This percentage is calculated from the base number of buffers specified when VTAM is started but does not include buffers from dynamic buffering in its calculations. Therefore, the default value of 60 percent is normally sufficient if you do not pre-allocate most of your I/O buffers at start time and if you use dynamic buffering. If you pre-allocate most of your I/O buffers at start time, you should consider overriding the VTAM default with a smaller percentage.

I/O Buffers and Application Program Data Transfer

Data that is transferred between a VTAM application program and another logical unit in the network passes through VTAM's I/O buffers. VTAM temporarily holds the data until an application program requests it or until it can be sent to a logical unit.

Figure 149 on page 592 shows how VTAM uses buffers for input operations.

Figure 150 on page 593 shows how VTAM uses buffers for output operations.

If you notice that your I/O buffers have expanded and you are running out of CSA, it is possible that an application program is not pacing data. Investigate the application programs that have a lot of traffic moving through the I/O buffers to determine whether pacing is being performed. For general information on pacing, see "Pacing and Flow Control" on page 159. For specific information on application program pacing, see "Application Program Pacing" on page 337.

Choosing Buffer Pool Specifications

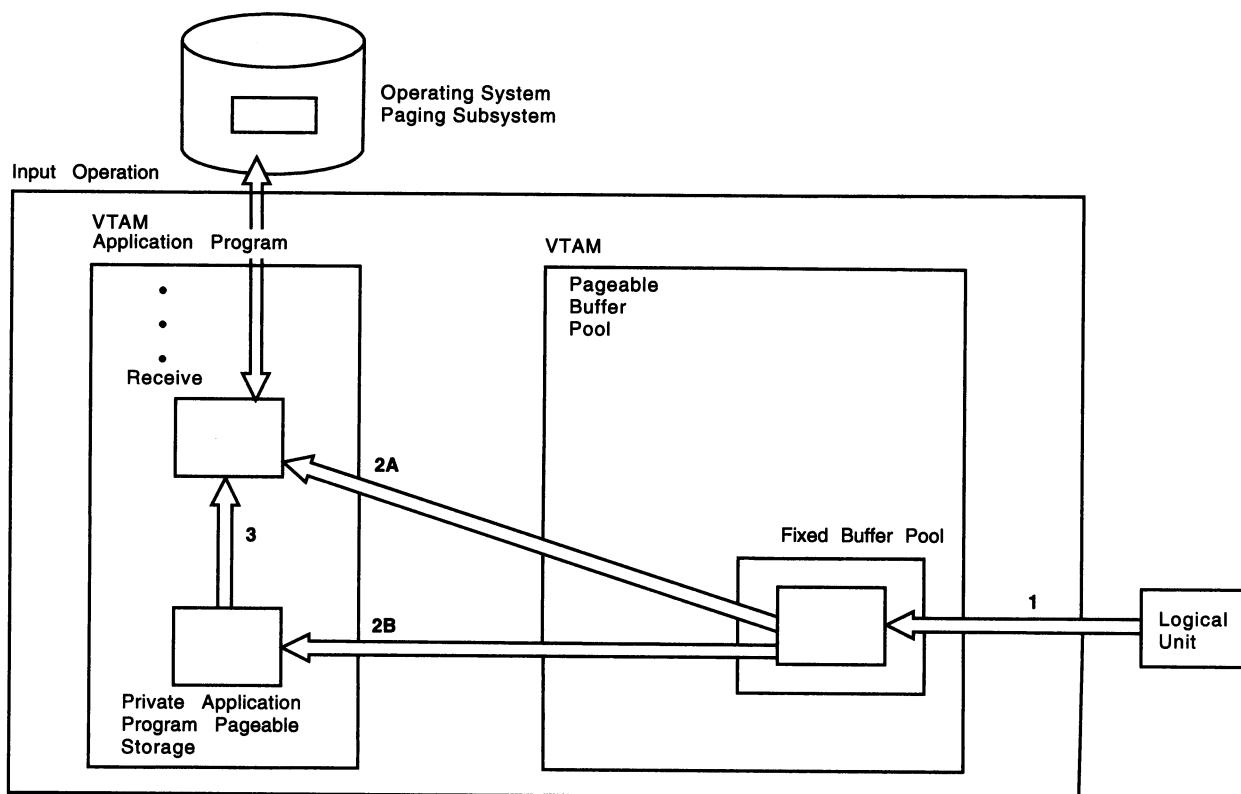


Figure 149. How VTAM Uses Input Buffers.

1. A logical unit in the network can send data to an application program independently of the application program's requesting the data. The data is placed in a buffer in the fixed buffer pool of data buffers.
2. One of the following occurs:
 - a. If the application program has an outstanding request for data from the logical unit, VTAM moves the data from the buffer in the fixed buffer pool to the specified application program data area. Later, data can be paged out of main storage by the operating system.
 - b. If the application program does not request data from the logical unit:
 - VM** VTAM sends the data to the application program's private storage.
 - MVS, VSE** VTAM places the data in a data space.
3. When the application program issues an input request, the operating system pages the data into main storage (if necessary) and VTAM moves the data to the specified application program's data area.

In general, an application program should be written so that at least one request for input is always outstanding. This reduces the chance of VTAM using excessive amounts of an application program's private storage.

VM You can code the MAXPVT operand on the APPL definition statement to limit how much of an application program's private storage VTAM can use.

HOT I/O Detection/Termination

VTAM has functions to both detect and terminate sessions deemed to be using excessive amounts of the IO buffer pool. The IO buffer pool is the storage pool that contains session data. Typically, these sessions are undergoing some sort of malfunction causing an unrestrained transmission of data into VTAM. A failure of VTAM to detect and terminate these out-of-control sessions can result in a VTAM or system outage due to CSA exhaustion. These sessions are commonly referred to as HOT or HOT I/O.

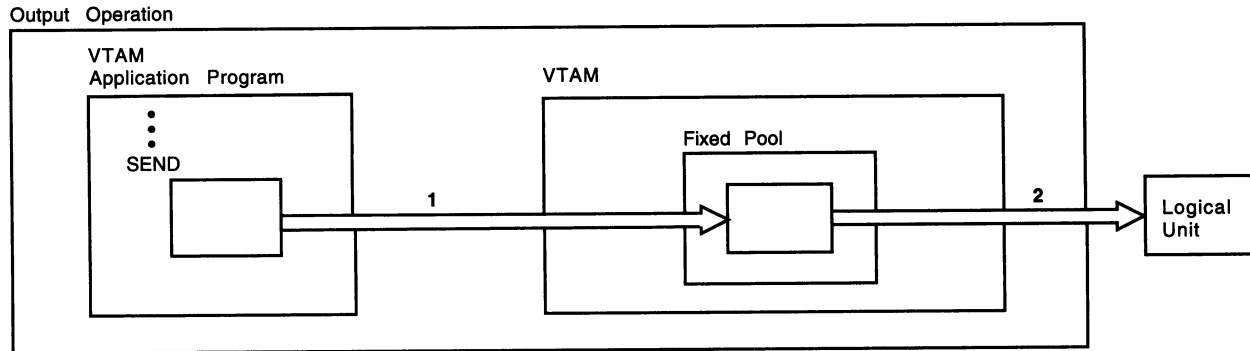


Figure 150. How VTAM Uses Output Buffers.

1. The application program requests data transmission. When VTAM has sufficient buffers in the fixed pool, it moves the data to the fixed pool.
2. VTAM sends the data to the NCP or the logical unit, freeing the buffer.

HOT I/O Session Detection

VTAM scans the entire IO buffer pool for sessions using excessive I/O buffers under two conditions:

1. When a successful expansion of the IO buffer pool brings the total size of the pool to 80% (or more) of the expansion limit specified (in the IOBUF statement). Note that this happens for each expansion in which the size of the pool is greater than the above mentioned threshold.
2. When an expansion of the IO buffer pool is unsuccessful (for any reason). In this case message IST154I is issued containing the expansion failure reason code. Note that if an expansion limit is not specified then this is the only condition in which a HOT I/O scan is performed.

When one of the above conditions occurs, VTAM scans the entire IO buffer pool looking for sessions using greater than 10% of the current size of the IO buffer pool. Any session found to be using greater than 10% of the pool will have a IST930I message issued for it. This message contains the session partner names as well as the percent of the pool in use by the session.

Specifying an expansion limit for the IO pool is advantageous in that VTAM can detect and correct HOT I/O conditions prior to a complete exhaustion of Common Service Area (CSA). There are disadvantages involved when an incorrect value has been specified. Consequently, it is recommended that the following guidelines be carefully evaluated so that a proper value may be specified.

If the expansion limit has been specified too low, one or more of the following may occur:

- VTAM uses extra CPU cycles doing HOT I/O scans. This primarily occurs when the expansion limit is specified so low that under normal workloads the size of the IO buffer pool is already greater than 80% of the expansion limit. The amount of CPU cycles that VTAM expends doing a HOT I/O scan is dependent on the size of the IO buffer pool.
- VTAM cannot expand the IO buffer pool. This occurs when the expansion limit has been specified so low that VTAM cannot expand the IO pool to accommodate a normal workload increase due to it reaching the expansion limit. VTAM may be adversely affected by a failure to obtain more IO buffers.

If the expansion limit has been specified too high, one or more of the following may occur:

- **MVS, VM** VTAM does not detect a HOT I/O condition until after CSA has been exhausted or limited by some other factor (such as the CSALIMIT start option). The expansion limit was specified so high that the size of the pool never reached the 80% threshold and consequently VTAM never scanned for HOT I/O until an expansion failed. This in effect works as if there is no expansion limit specified. Consequently, VTAM and the system may be adversely affected by this condition.
- **MVS, VM** A VTAM or system outage occurs due to an exhaustion of CSA. This primarily occurs when VTAM's CSA use is not also limited with the CSALIMIT parameter. If VTAM is not limited in the amount of CSA it can acquire and a HOT I/O situation occurs, VTAM may use so much of CSA that the system (GCS or MVS) is adversely affected.
- **VSE** The above MVS, VM items are also true for VSE, except that CSA should be SGA and CSALIMIT should be SGALIMIT for VSE.

HOT I/O Session Termination

Once one or more sessions have been identified as using greater than 10% of the pool, VTAM will then attempt to stop the buffer pool expansion with one of the following actions:

- Inactivate the LU, when the session type is SSCP-LU.
- Inactivate the PU, when the session type is SSCP-PU.
- Terminate all sessions between the two logical units, when the session type is LU-LU.

The above action will be taken if all of the following criteria are met:

- The percentage of the pool in use by the session is greater than or equal to the HOT I/O session termination threshold as specified by the HOTIOTRM start option.
- Neither session partner is the SSCP.
- The subarea of one or both session partners must be the subarea in which the HOT I/O is being detected.
- When the session type is SSCP-PU, the PU is not an NCP.

If all of the above conditions are true and it is a LU-LU session, VTAM terminates all sessions between the two session partners. It then issues message IST1099I indicating that the sessions have been terminated.

If all of the above conditions are true and it is a SSCP-LU or SSCP-PU session, VTAM inactivates the LU or PU. It then issues message IST1098I indicating that the LU or PU has been inactivated.

Controlling NCP Slowdown

To alleviate the slowdown condition, you can take one or more of the following actions:

- Select a buffer size that makes more efficient use of NCP buffers.
- Install more memory in the communication controller.

- Transfer inbound data to the CPU more quickly by decreasing DELAY.
- Transfer outbound data to the network more quickly by increasing line speed.

Note: Many systems have much more outbound data than inbound. Therefore, increasing line speed can cause more data to be transferred out of the NCP through the line. This reduces buffer use.

- If the NCP slowdown can be attributed to any given session or sessions, session pacing can be used to slow down the sessions causing the congestion.

NCP clears slowdown conditions by accepting outbound data from the CPU more slowly and by accepting data from the network more slowly. It stops polling and requests the host to stop writing data. Slowdowns are reduced in the following manner: a device is not polled if a PIU READ puts the NCP in a slowdown condition. The value specified for MAXDATA can control the speed with which data is placed on intermediate routing node (IRN) links.

The following actions can be taken to alleviate slowdown for local SNA controllers:

- Increase the value of MAXBFRU. This allows the controller to send more data on every read channel program.
- Use session pacing to control the amount of data VTAM sends to the controller. Specify lower pacing values for primary-to-secondary pacing. This limits the number of PIUs that are sent to the controller by VTAM.

Maximizing Coattailing

To maximize coattailing, you need to transfer more data either inbound to the host or outbound without generating an attention interrupt. If more messages are transferred in or out of the host than the number of READ or WRITE channel programs issued, coattailing is taking place. The effect is to transfer more data in a single channel I/O operation. Coattailing can provide greater throughput between a host and an SNA controller; however, response time can increase.

The amount of coattailing that you can achieve is directly related to the message traffic in your network. However, you can influence the amount of data that is available for a data transfer operation.

Controlling Outbound Coattailing

By controlling the channel delay timer for VTAM and the attention interrupt delay timer, you can affect outbound coattailing in your system.

VTAM saves PIUs for transfer over a channel until one of the following events occurs:

- Data flowing at transmission priority 2 (TP2) is available.
- A virtual route pacing response must be transmitted.
- An attention interrupt from an NCP or an SNA cluster controller (for example, the 3174) occurs.
- The PIU queue for VTAM reaches its limit.

For each VTAM channel delay interval, VTAM calculates the queue depth

(QDPTH tuning statistic) by multiplying the number of PIUs transferred during the channel delay interval by a percentage (currently 75%).

- The channel delay timer or the backup 3-second timer expires.

Controlling Inbound Coattailing

To affect inbound coattailing, delay the VTAM WRITE operation. This is effective because VTAM normally reads data after transferring data to the SNA controller or to another VTAM. By delaying VTAM longer before it performs a WRITE operation, more data can potentially accumulate in the SNA controller or other VTAM host, which can be read in the same I/O operation as the WRITE channel program. You can control the time interval during which VTAM buffers low priority data before transferring it over the channel to another VTAM, NCP, or SNA cluster controller by coding the DELAY operand on the appropriate VTAM definition statement.

DELAY Operand

When defining a channel, you should consider the effects of the DELAY operand on the LINE definition statement. This operand specifies the maximum amount of time an outbound PIU with transmission priority 0 or 1 should be queued before being transmitted over the channel. When the PIU is eventually sent, it and the PIUs following it that have also been queued are sent as a block.

For VTAM, the DELAY operand specifies the maximum amount of time that VTAM queues an outbound PIU with transmission priority 0 or 1 before transmitting it over the channel. When the PIU is eventually sent, it and the PIUs following it that have also been queued are sent as a block.

When you specify DELAY=0, VTAM sends each PIU over the channel as it arrives on the queue. When a nonzero value for DELAY is specified, the blocking of low-priority PIUs can save host processor instructions. However, a high channel delay time can cause increased response time because some of the PIUs are delayed before being sent.

Occasional bursts affect throughput only for the next second or two. However, a continuing series of bursts can severely affect capacity if there is little non-channel-to-channel activity. A few sessions of interactive traffic are more likely to accentuate this condition than a large number, when traffic on the channel is likely to be more randomly distributed.

You might sometimes want to increase the DELAY operand value over that of the default to do the following:

- Decrease host processor overhead when response time is less important.
- Take advantage of a smoothing effect on the queuing that can occur with larger values, an effect that can be useful for application programs with sessions conducted in bursts.

You can specify different DELAY values for different channel-attached devices. The following are general guidelines for picking an appropriate DELAY value for a particular channel-attached device.

If you want to experiment with DELAY values greater than the default, you can activate in succession a series of major nodes with increasing values beyond the default.

MAXBFRU Operand

For channel-to-channel connections, the MAXBFRU operand defines a constant sized buffer in increments of 4K pages. The value does not fluctuate throughout the life of the connection. This buffer must be defined as large as the largest PIU to cross the channel-to-channel connection. Also, if a nonzero DELAY value is chosen and you want to transfer multiple RUs in a single I/O operation, the MAXBFRU value must also be large enough to handle multiple RUs.

After the data transfer operation is complete, the PIUs are deblocked into VTAM buffers as defined in the IOBUF start option. The VTAM buffers are allocated in 4K increments so you might have to allocate more I/O buffers. If you are using dynamic buffer expansion, you should monitor the I/O buffers pool to ensure that there is not excessive buffer pool expansion and contraction because of these channel operations.

The maximum PIU size across a connection is defined by MAXBFRU times IOBUF size.

If all LU-LU sessions end in a physical unit that provides segment assembly, path information unit (PIU) segmentation occurs across the link. If any LU-LU sessions end in a physical unit that does not provide segment assembly, the result of MAXBFRU times IOBUF size must be large enough to handle RU sizes for those sessions. If the LU-LU session cannot have segments, either increase the value of the MAXBFRU operand or the I/O buffer sizes, or adjust the RU size in the MODEENT macroinstruction for the logical unit.

Coattailing for SNA Controllers

The amount of coattailing of messages that occurs between a host and a channel-attached NCP or SNA controller is affected by the following:

- I/O buffer size
- The number of buffers allocated to receive incoming messages at the host
- The amount of time that VTAM and the channel-attached device allow to elapse before sending buffered messages.

The initial amount of storage you allocate for the I/O buffer, as well as the *xpanno* value you specify, depends on the *bufsize* specification. This, in turn, depends on the UNITSZ value specified in both the NCP HOST definition statement and the buffer pool start option for the I/O buffer. In the case of channel-attached cluster controllers and channel-to-channel adapters, *bufsize* depends only on the buffer pool start option for the I/O buffer. The value in the VTAM start option is used by both VTAM and the NCP when they exchange information. The *bufsize* value entered as part of the buffer pool start option for the I/O buffer is interpreted by VTAM as the UNITSZ value used. The I/O buffer size, as it appears on the console after a DISPLAY BFRUSE operator command, is greater than the *bufsize* that was specified in the buffer pool start option because VTAM adds control information to each I/O buffer.

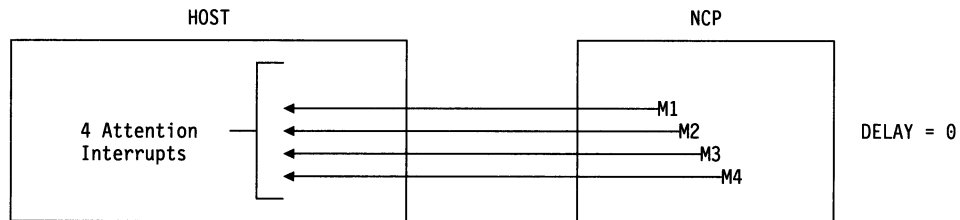
Guidelines for Setting DELAY

To specify the amount of time that elapses before VTAM or NCP sends buffered data, use the DELAY operand on NCP and VTAM definition statements. The DELAY operand on the NCP BUILD definition statement controls how long the NCP buffers data; it is the elapsed time between the receipt of the first inbound message and the presentation of an attention interrupt to the host. This capability can affect your strategy for increasing coattailing use of the channel.

The DELAY operand that controls how long VTAM buffers data can be specified on the following:

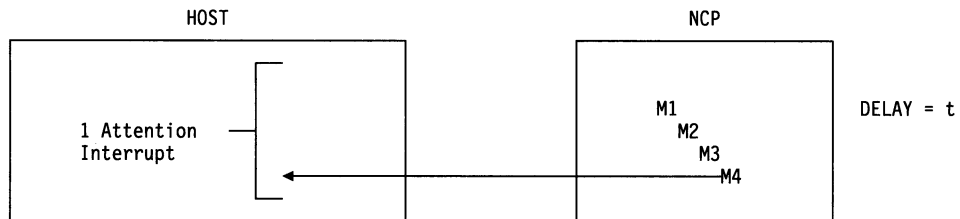
- PCCU definition statement for NCPs not defined in a channel-attachment major node
- GROUP, LINE, or PU definition statements for NCPs defined in a channel-attachment major node
- PU definition statement for local (channel-attached) SNA controllers.

The three examples in Figure 151 illustrate the use of the DELAY operand.



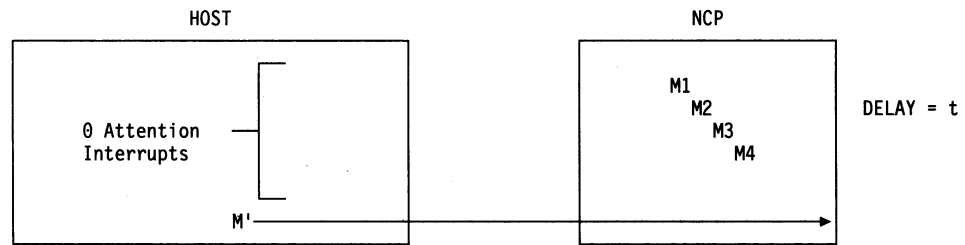
In the first example, four messages have come in over a period of time. Because DELAY is 0, each message has resulted in an attention interrupt being presented to the host. In all, four attention interrupts have been presented, and no coattailing has taken place.

Figure 151 (Part 1 of 3). Effect of DELAY Time on Coattailing



In the second example, DELAY is equal to t . This time period is large enough to allow four messages to arrive at the NCP. The time t has expired, and the NCP presents an attention interrupt. VTAM reads all four messages, having received only one attention interrupt.

Figure 151 (Part 2 of 3). Effect of DELAY Time on Coattailing



In the third example, DELAY is still equal to t , but before t expires and after four messages arrive at the NCP, an outbound message (M') is sent from VTAM to NCP. In this case, four messages are sent to the host with no attention interrupts generated. This is the ideal case, in terms of saving instructions executed in the host.

Figure 151 (Part 3 of 3). Effect of DELAY Time on Coattailing

For NCP, the DELAY operand allows time for more than one message to arrive in the NCP before an attention is presented to the host.

The following events cause PIUs on the queue to be sent before the DELAY time elapses:

- VTAM receives an attention interrupt from NCP.
- VTAM or NCP reaches the MAXBFRU limit.
- VTAM sends virtual route pacing responses.
- VTAM sends transmission priority 2 traffic.
- VTAM sends APPN network priority or highest priority traffic (to a local SNA PU).

The default value for the DELAY operand is 0.2 second (200 ms) for SNA controllers. These are appropriate values for most networks. However, under the following conditions, you might need to change the value to 0:

- You have a low rate of transactions that use the channel connection, and response time for interactive transactions is important.
- You have a low rate of file transfer with little other activity, but elapsed time for file transfer is important.
- You have channel traffic with frequent bursts of SENDs (such as from an application program that makes inquiries of the other host in rapid succession).

To analyze inbound coattailing for channel-attached SNA communication or cluster controllers, compare the number of times that NCP signals VTAM that it has data to send (ATTN tuning statistic) to the number of times that VTAM issues a channel READ program to read data (CHRD tuning statistic). A less accurate method is to divide the number of inbound PIUs transferred by NCP (IPIU tuning statistic) by the number of read channel programs issued by VTAM (CHRD). This calculation gives you the average number of PIUs read from the channel-attached device with each channel program, which is an indicator of coattailing.

Note: For a 3274 to clear its buffers after a data transfer operation, the 3274 signals to VTAM that it has more data to send even though there is no data. This causes the RDATN tuning statistic to equal the CHRD tuning statistic, and because RDATN is included in the ATTN tuning statistic, the ATTN value is normally double the CHRD value.

To analyze outbound coattailing for channel-attached SNA communication or cluster controllers, divide the number of outbound PIUs (OPIU tuning statistic) by the number of WRITE channel programs VTAM issues (CHWR tuning statistic). This calculation gives you the average number of PIUs transferred by VTAM with each channel write program, which is a rough estimate of the effectiveness of coattailing.

Note: The DELAY specification for a 3174 is a customization option. However, even if you set the DELAY customization option for a 3174 that is performing a token-ring gateway function, the option is ignored. The DELAY operand in VTAM should also be set to 0 for a 3174 that is a token-ring gateway channel-attached cluster controller.

Specifying a nonzero attention DELAY value to activate attention DELAY does not automatically ensure coattailing. To ensure coattailing, all of the following must be true:

- The installation must have a transaction rate high enough to allow more than one message to accumulate in the DELAY time period in the NCP or the host.
- The MAXBFRU value must be large enough to allow more than one message to be sent to the host.
- The host processing speed has an effect on coattailing; the faster host processors reduce coattailing slightly. An attention interrupt occurs when MAXBFRU is reached or when the DELAY time period expires.

Begin by setting DELAY equal to or greater than 0.2. A 0.2-second delay has little effect on response time, but if the traffic speed is approximately 15 transactions per second or greater, coattailing occurs.

Guidelines for Setting MAXBFRU

To specify the number of VTAM I/O buffers allocated for inbound data transfer, use the MAXBFRU operand on the NCP HOST definition statement and on the VTAM GROUP or LINE definition statements for channel-attached NCPs or SNA controllers. When VTAM creates a read channel program, it must always have MAXBFRU number of I/O buffers available for reading from the channel-attached device. VTAM always attempts to hold in reserve MAXBFRU number of I/O buffers for the next read from the channel-attached device.

Set MAXBFRU so that MAXBFRU times UNITSZ is the largest PIU the NCP can send to VTAM. When you set MAXBFRU and UNITSZ, remember that MAXBFRU times UNITSZ must be greater than or equal to TRANSFR times BFRS minus 18 ($\text{MAXBFRU} \times \text{UNITSZ} \geq \text{TRANSFR} \times \text{BFRS} - 18$).

In addition, choose MAXBFRU so that more messages can be coattailed. The MAXBFRU value determines when an attention interrupt is signalled to the host. MAXBFRU=5 is a good starting point. There should be enough buffers available to transfer all the messages likely to arrive in the DELAY time period. (The use of MAXBFRU=5 assumes one inbound message fits in one I/O buffer. For more details on I/O buffer size, see the following discussion.) You can use tuning statistics to verify that MAXBFRU is sufficient for coattailing. Using the example in Figure 154 on page 602, for instance, MAXBFRU could have been set to 2. Instead, it was set to 6 to maximize coattailing.

The number of buffers required for a message (RU) is determined by the following formula:

$$(29 + \text{RU size}) \div \text{bufsize}$$

The number 29 is obtained by adding the size of the transmission header (26) to the size of the request/response header (3). The *bufsize* is the buffer pool value you use to start VTAM. If you are unsure about this value, look at your start options for the IOBUF buffer size specification. This value should be greater than or equal to the UNITSZ operand for any NCPs you are using.

The MAXBFRU value you use should be greater than or equal to the number of buffers required for the largest message (RU) your system receives. This minimizes the number of attention interrupts needed to process large messages. Use the preceding formula to calculate the lower limit for MAXBFRU, given the size of the largest message.

Guidelines for Setting I/O Buffer Size

Choosing the ideal I/O buffer size helps minimize host storage requirements and host processor usage.

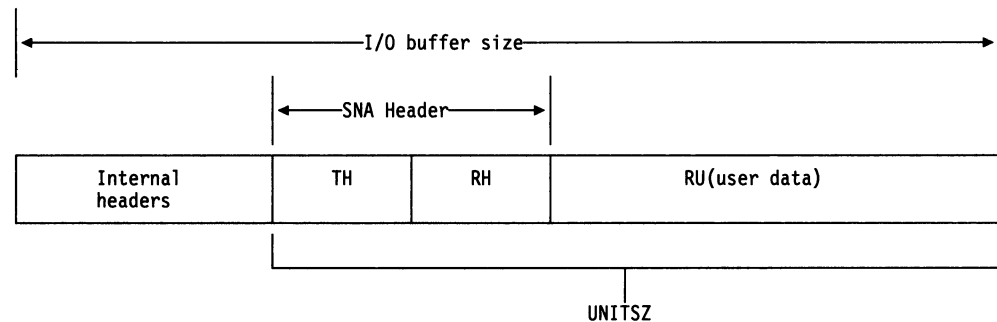
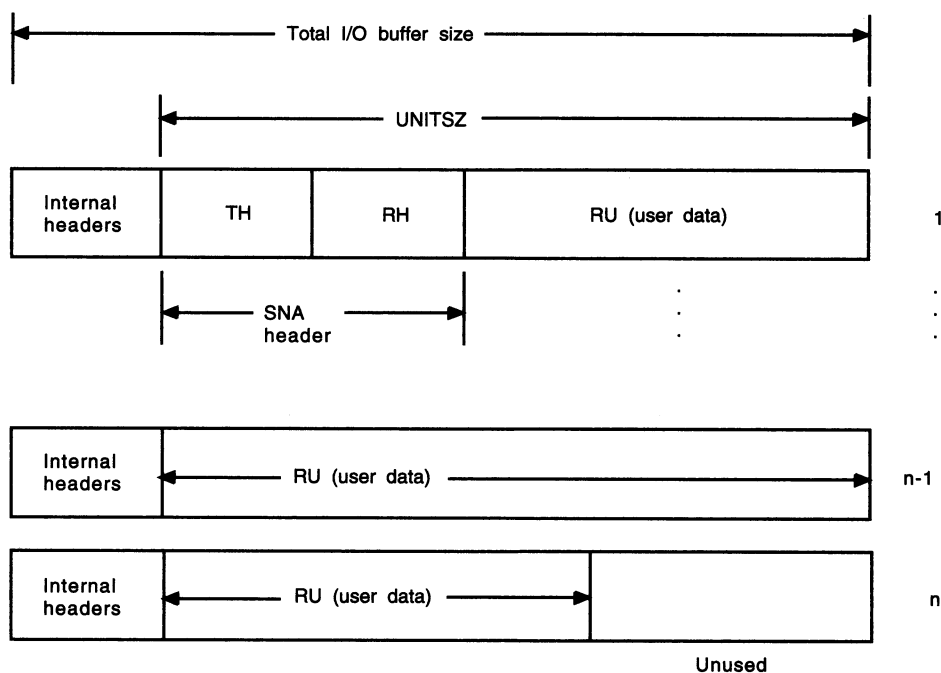


Figure 152. General I/O Buffer Format

Figure 152 illustrates UNITSZ. UNITSZ is not the size of an I/O buffer; it is a portion of the buffer size. It includes an SNA header field. In the remaining discussion, the I/O buffer size is referred to as the UNITSZ. (Note that one buffer size is used for both inbound and outbound data transfers.)

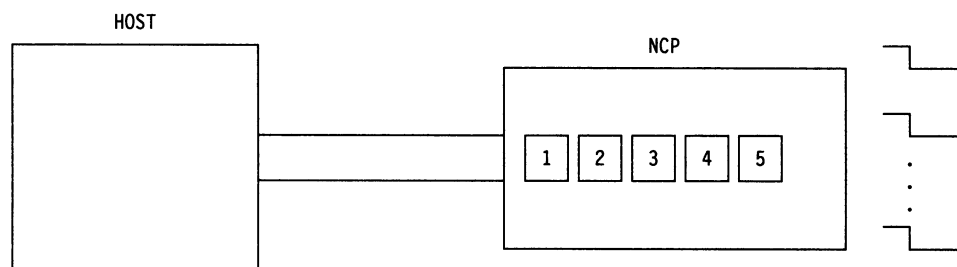


Note: Multiple I/O buffers are used where the length of outbound data is greater than UNITSZ minus the SNA header.

Note: A similar concept applies to inbound data.

Figure 153. Using Multiple I/O Buffers to Transfer Single Message

Figure 153 illustrates how a message that is larger than UNITSZ spans more than one buffer. The message can be greater than UNITSZ. If it is, VTAM uses more than one buffer to contain the message. The unused space in the last buffer is not allocated to the next message.



MAXBFRU=6, DELAY=t, five messages arriving; each message needs two buffers.

- 1 ← I/O attention interrupt
- 2 ← Three messages transferred
- 3 ← Attention bit set
- 4 ← Two messages transferred

Figure 154. Multiple-Buffer Considerations

Figure 154 assumes that MAXBFRU=6 and DELAY=t and that there are five messages arriving at the NCP before the time t expires. However, it also assumes that each message requires two I/O buffers. After three messages are received, the MAXBFRU limit is reached, and the NCP signals the host with an attention inter-

rupt. The three messages are transferred, but two more remain. Therefore, the NCP signals VTAM that there is more data to read (RDATN tuning statistic). If the RDATN tuning statistic is high, NCP has more data to transfer to VTAM than VTAM has allocated buffers to process. Therefore, you can increase the MAXBFRU specification, thus allocating more VTAM buffers to reduce this condition.

However, coattailing has taken place; two messages have been transferred without having generated an attention interrupt. VTAM creates another channel program to read the remaining messages. Again, coattailing has occurred, because two messages have been transferred and only one attention interrupt has occurred. However, two I/O operations have been used where the data transfer might have been accomplished with only one. If UNITSZ were large enough to contain an entire message or if MAXBFRU were equal to 10, the extra I/O operation could have been avoided.

Guidelines for Setting UNITSZ

To choose the best UNITSZ value, first determine the average size of inbound and outbound messages. Then, compare them to each other, and set the UNITSZ equal to the larger of the inbound or outbound message sizes.

Note: The message size is not determined by VTAM. You can determine the average message size by examining your application programs (for example, CICS and IMS). You can also use the NetView Performance Monitor to analyze the average message size for your network.

Suppose the inbound size is much smaller than the outbound size. Choose a UNITSZ at least large enough to contain one complete inbound message. However, the UNITSZ should be larger than the inbound message so that relatively few buffers are needed for the outbound message. Because the extra buffers increase path length, it is better not to specify too many buffers.

As another example, consider the case where the inbound message size is much greater than the outbound size. The process is similar to the previous one, except that the outbound message is the guide for choosing UNITSZ. Choose a UNITSZ that is greater than or equal to the outbound message size so that relatively few buffers are needed to contain one inbound message.

Where practical, choose UNITSZ so that no more than five to seven buffers are needed for transferring one message, whether it is inbound or outbound.

Coattailing for Channel-to-Channel Operations

VTAM channel-to-channel I/O operations, which are similar to SNA controller operations, are impacted by factors such as VTAM I/O buffer size and the channel delay and maximum number of buffer pages on the CTC definitions. The following topics discuss variables that can affect VTAM channel-to-channel performance.

Guidelines for Setting DELAY

To select a DELAY operand value, use the following procedure:

1. During periods of the day when it is most likely that you are experiencing performance problems, turn on the tuning statistics. For channel-to-channel attachments, turn on tuning statistics at each processor using the following operator command:

```
F NET,TNSTAT,CNSL,TIME=1
```

2. For each recording of the tuning statistics, check the amount of coattailing that is occurring. For channel-to-channel connections, compare the TIMERS and CHNRM values of the channel-attachment major node associated with each host processor. The desirable TIMERS value is 0; an occasional nonzero value is acceptable.
3. For channel-to-channel attachments, if the TIMERS value in any tuning statistics record is too large, deactivate the channel-attachment major node at each host processor, and activate a previously defined major node in which DELAY=0 has been specified on the LINE definition statement.

For other channel attachments, if the average number of PIUs seems too high (thereby indicating possible response time problems), you can activate alternate definitions with a lower DELAY value.

The following events cause PIUs on the queue to be sent before the DELAY time elapses:

- VTAM reaches the MAXBFRU limit.
- VTAM sends virtual route pacing responses.
- VTAM sends transmission priority 2 traffic.
- VTAM reaches the QDPATH value.

The default value for the DELAY operand is 0.1 second (100 ms), which is an appropriate value for most networks. However, under the following conditions, you might need to change the value to 0:

- A low rate of transactions that use the channel-to-channel connection. A rate of under five transactions per second is considered low.
- Channel-to-channel traffic with frequent bursts of SENDs, such as from an application program that makes inquiries of the other host in rapid succession.

You can also analyze the average number of bytes transferred per I/O operation by dividing the number of total number of bytes transferred (RDBUF) by the number of READ channel programs (CHNRM).

Guidelines for Setting I/O Buffer Size

Because data is transferred in the channel-to-channel buffers, increasing or decreasing the I/O buffer size has no effect on the channel program size.

Virtual Route Window Sizes

The default minimum window size might be too small for a VTAM channel-to-channel virtual route. VTAM increases the window size only when the virtual route pacing response is not returned fast enough to prevent one window's worth of PIUs from being queued up on the virtual route (a HELD condition). A channel-to-channel route is so fast that the virtual route pacing responses are turned around fast enough to prevent HELD conditions from occurring very often. Because of this, the current virtual route window size tends not to increase, but to stay very close to the minimum VR window size.

The smaller the VR window size, the more virtual route pacing responses flow over the channel. This can cause higher CPU utilization than necessary.

Setting a higher minimum window size reduces the number of virtual route pacing responses and helps CPU utilization and throughput. For channel-to-channel one-hop routes, the default minimum VR window size is 1. The recommended

value is 15. The recommended maximum virtual route pacing window size value is 50 for this environment.

If VTAM is expanding its IOBUF buffer pool to service a virtual route, the lack of VTAM buffers is considered major congestion for the route. Therefore, VTAM decreases the virtual route window size to the minimum for all VRs that it services. You should monitor this VTAM buffer pool to ensure that there is not excessive expansion and contraction if you are using dynamic buffering.

You can also use VTAM tuning statistics to analyze a virtual route's impact on the channel-to-channel connection. A virtual route that uses transmission priority 2 and traverses the channel-to-channel connection causes VTAM to immediately schedule the data transfer operation. A virtual route pacing response is also high priority traffic, which has the same effect. The PRI tuning statistic in VTAM indicates the number of times that a VTAM channel program is started to transfer this high priority data (TP2 or VR pacing response) to the other VTAM host. If this number is high and the channel-to-channel connection is not used extensively for TP2 traffic, the minimum virtual route pacing window size is probably too small. Also, the higher this number is in relation to the sum of TIMERS, QDPTH, and BUFCAP, the less outbound coattailing occurs.

For more information about pacing, see “Pacing and Flow Control” on page 159.

Appendixes

Appendix A. TSO/VTAM	611
Defining TSO to VTAM	611
Defining the TCAS Application to VTAM	612
Defining TSO/VTAM Session Parameters	614
Defining an Interpret Table for Compatible Logons	620
Defining TSO to MVS	620
Writing a Procedure to Start TSO/VTAM Time Sharing	620
Creating a TSOKEY00 SYS1.PARMLIB Member	620
Defining TCAS Program Properties	620
Implementing TSO/VTAM	621
Translation Tables	621
Coding TSO/VTAM Exit Routines	621
Security	621
Performance	622
3270 Large Screen Considerations	622
TSO Considerations	623
National Language Support for TSO/VTAM End-User Messages	623
Operating VTAM under TSO	624
How TSO/VTAM Differs from TSO through TCAM	624
 Appendix B. VM/SNA Console Support	627
Preparing to Start VSCS	627
Defining VSCS as a VTAM Application Program	627
Authorizing the VSCS Virtual Machine	628
Updating the Directory Entry for the VSCS Virtual Machine	628
Preparing the GCS Environment to Start VSCS	629
Tailoring VSCS	630
Changing VSCS Start Options	630
Session Parameters for Devices: Logon Mode Tables	631
LOGON and LOGOFF Commands: USS Tables	632
Data Streams: VSCS Data Manipulation Exit Routines	632
Protecting Confidential Data	633
VSCS Start Options	633
Controlling VSCS Storage Management	633
Controlling VSCS Dumps	635
Controlling VSCS Performance	636
Controlling VSCS SCIP Recovery	638
Defining Your Own Set of VSCS Start Options	639
Using VSCS Diagnostic Tools	640
Tracing Commands	640
Display Command	641
LU Dump Command	641
Storage Command	641
Buffer Commands	641
Query Command	641
Halting VSCS	642
Additional VSCS Considerations	642
Printers	642
Terminal Logons	643
Terminal Logoffs	644

VM Logo Usage	645
Accounting Information	647
Appendix C. Tuning VM for VTAM	649
SET RESERVE	649
DIAG98	649
Page Locking	649
XEDIT SET REMOTE	650
SET QUICKDSP	650
SET SHARE RELATIVE 10000	651
Appendix D. Logon Manager	653
How the Logon Manager Operates	653
Installing the Logon Manager	654
Starting the Logon Manager	654
Defining the Logon Manager	655
Sample Logon Manager Configuration	655
Defining the Logon Manager and TPF Applications to VTAM	656
Defining the Logon Manager Configuration	657
Monitoring Logon Manager Resources	661
Halting the Logon Manager	661
Appendix E. Cryptographic Keys	663
Filing SLU Keys for Single-Domain Cryptographic Sessions	663
Single-Domain Cryptographic Sessions That Use PCF/CUSP	663
Single-Domain Cryptographic Sessions That Use ICSF/MVS	664
Filing CDRM Keys for Cross-Domain Cryptographic Sessions	665
Cross-Domain Cryptographic Sessions Where Both Hosts Use PCF/CUSP	665
Cross-Domain Cryptographic Sessions Where Both Hosts Use ICSF/MVS	667
Cross-Domain Cryptographic Sessions Where the Hosts Use Different Cryptographic Services	669
Changing a Logical Unit's Cryptographic Capability	671
Appendix F. Command Lists: Dynamic Configuration of Channel-Attached Devices	673
ISTDINFO: VTAM Device Information Services	673
Dependencies and Restrictions	673
Output Variable	674
Output Tokens	674
ISTDEFIN: VTAM Device Information Services	677
Dependencies and Restrictions	677
Output Variable	678
Output Tokens	678
Appendix G. Message Translation Using the MVS Message Service	681
Overview of MMS Support	681
Internal Translation	681
Selecting Internal Translation	681
Defining Messages for Internal Translation	682
External Translation	682
Selecting External Translation	682
Defining Messages for External Translation	683
Skeleton File Use	684

Appendix H. Forcing an APPN Route in a VTAM Network	685
Appendix I. Border Node Connection Types	691
Appendix J. VTAM Restricted Materials	695
Appendix K. VTAM AnyNet Feature	697
Overview of Multiprotocol Transport Networking (MPTN)	697
Peer and Transport Relationships	697
MPTN Architecture Functions	698
MPTN Transport Services	699
MPTN Network Management	702
Overview of the VTAM AnyNet Feature	703
SNA over TCP/IP	703
Sockets over SNA	704
Sockets over SNA Gateway	704
Appendix L. Communications Manager/2's DLUR Support	705
Downloading the DLUR Files	705
Unpacking the DLUR Files into Communications Manager/2 Subdirectories	706
Considerations for VTAM Switched Major Node Definitions for DLUR	708
Additions to Communications Manager/2 Configuration Commands for DLUR	708
Addition to DEFINE_LOGICAL_LINK	709
DEFINE_DEPENDENT_LU_SERVER	709
Sample Communications Manager/2 NDF File for DLUR	711
Sample Communications Manager/2 RSP File for DLUR	712

Appendix A. TSO/VTAM (MVS)

TSO/VTAM provides the capability of using TSO through VTAM. TSO is a standard feature in MVS that provides conversational time sharing.

TSO/VTAM supports the following terminals:

- IBM 3270 Information Display System
- IBM 3290 Information Display System
- IBM 3767 Communication Terminal
- IBM 3770 Data Communication System
- IBM 8775 Display Terminal
- IBM Displaywriter
- IBM 5550 (as 3270) single-byte character set (SBCS) and double-byte character set (DBCS) modes.

TSO/VTAM can also support the following non-SNA terminals in conjunction with the IBM Network Terminal Option licensed program:

- IBM 2741 Communication Terminal
- CPT-TWX Terminal
- World Trade Telegraph (WTTY).

The basic elements of TSO/VTAM are the terminal control address space (TCAS) and the VTAM terminal I/O coordinator (VTIOC). TCAS accepts logons from TSO/VTAM users and creates an address space for each user. VTIOC is the interface between TSO and VTAM; it coordinates data flow. TCAS and VTIOC, together with existing TSO components such as the LOGON scheduler, the application programs, the terminal monitor program, and the TSO service routines, make up a TSO/VTAM time-sharing system.

The user logon to the TCAS address space is passed to the TSO user address space without initially establishing a session between the TCAS address space and the TSO terminal. Instead, the session is established between the TSO terminal and the TSO user address space, avoiding session establishment and termination with the TCAS address space and avoiding your receiving message USSMSG7.

TSO/VTAM and TSO through TCAM can reside in the same host processor.

Defining TSO to VTAM

To use TSO/VTAM with VTAM, define the following to VTAM:

- TCAS and each TSO user
- TSO/VTAM session parameters.

For compatible logons, you also need to create and define to VTAM an interpret table.

Defining the TCAS Application to VTAM

An APPL definition statement defines an application program to VTAM. Because TCAS and each TSO user are VTAM application programs, you must code APPL definition statements for them and put the definition statements in SYS1.VTAMLST.

You can reduce the number of operands you need to code on the APPL definition statements for each TSO user by using the GROUP definition statement under the APPL major node. All of the operands on the APPL definition statement except ACBNAME can be coded on the GROUP definition statement and allowed to sift down. You only need to code the unique name on the APPL definition statements for the TSO users if you use the GROUP definition statement to specify all the other operands.

Note: To avoid obtaining unnecessary CSA storage for TSO users, you should define TSO application programs to VTAM with EAS=1; do not use the default. TSO/VTAM obtains CSA storage independently of CSALIMIT.

Single-Domain Network

In a single-domain network in which there are no overriding network naming conventions, you can use the following technique to code the APPL definition statements for TCAS and for the TSO users:

Code the following APPL definition statement for TCAS:

```
tsoa      APPL  ACBNAME=TSO,PRCT=password,
           AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1
```

The label *tsoa* is a unique network name. This name can be up to 8 characters in length, but should be kept to less than eight so that it can be a prefix for the names on the subordinate TSO APPL definition statements.

Code as many APPL definition statements, in the following format, as there will be users logged on to TSO/VTAM at one time:

```
TSOannnn APPL  ACBNAME=TSOnnnn,PRCT=password,
           AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1
```

The label *tsoannnn* is a unique network name, which can be up to eight characters in length. It can be helpful to use the network name for TCAS as a prefix in the network name when coding APPL definition statements for each terminal logged on to the TSO/VTAM in this VTAM host concurrently. For the *nnnn* suffix, use sequential decimal integers starting with 0001. For the ACBNAME, the *nnnn* suffix is a decimal integer; the numbering must start with 0001 and be sequential.

Note: It is recommended that session pacing should not be used; NVPACE should be coded. However, in situations where pacing is necessary, then code a high value such as VPACING=20.

You need to use the same *password* for TCAS and each TSO/VTAM user. You must code a different application program name, in the form *TSOnnnn*, for each user. The *nnnn* suffix is a decimal integer; the numbering must start with 0001 and must be sequential. NOACQ and NOPO need not be coded; they are default values.

Multiple-Domain Network

For TSO sessions to be established in a multiple-domain network, you must code CDRSC definition statements for each TSO/VTAM application program unless you specify SSCPDYN=YES. Code the CDRSC definition statements in every VTAM host owning logical units that will log on to the application program.

In the VTAM host containing the TSO subsystem, it is not necessary to define the LUs in the other domains that issue logons to TSO/VTAM if dynamic CDRSC definition is authorized. Also, the manager in the domain of the SLU can be authorized to create CDRSCs for the TSO application program associated with the TSO user.

Code the following CDRM definition statements in each domain to avoid defining both the SLUs that log on to TSO/VTAM as CDRSCs in the TSO/VTAM domain and the TSO user application programs as CDRSCs in the SLU's domain.

```
name1    CDRM  CDRDYN=YES,CDRSC=OPT,...
```

```
name2    CDRM  CDRDYN=YES,CDRSC=OPT,...
```

The CDRDYN operand determines whether a CDRM is authorized to dynamically create a CDRSC representing a cross-domain LU when a logon request is received from the LU (through an Initiate from the CDRM managing the LU).

The CDRSC operand determines whether the dynamic creation of CDRSC definitions is permissible when a logon request is received from the CDRSC manager identified by this CDRM definition statement.

In a multiple-domain network where TSO/VTAM is run in more than one domain, you must have special definition statements with unique names for TCAS and the terminals in your domain and interacting domains. Code the following APPL definition statement for TCAS in your domain:

```
tsoa      APPL  ACBNAME=TSO,PRCT=password,
                AUTH=(NOACQ,PASS,NVPAGE,
                TSO,NOP0),EAS=1
```

The label *tsoa* is a unique network name. This name can be up to 8 characters in length, but should be kept to less than eight so that it can be a prefix for the names on the subordinate TSO APPL definition statements.

Code as many APPL definition statements in the following format as the maximum number of sessions that will be established with TSO/VTAM in your domain at one time:

```
tsoannnn APPL  ACBNAME=TSOnnnn,PRCT=password,
                AUTH=(NOACQ,PASS,NVPAGE,
                TSO,NOP0),EAS=1
```

The label *tsoannnn* is a unique network name, which can be up to eight characters in length. It can be helpful to use the network name for TCAS as a prefix in the network name when coding APPL definition statements for each terminal logged on to the TSO/VTAM in this VTAM host concurrently. For the *nnnn* suffix, use sequential decimal integers starting with 0001. For the ACBNAME, the *nnnn* suffix is a decimal integer; the numbering must start with 0001 and be sequential.

Code the following definition statement for each TCAS in another domain with which an LU in your domain communicates:

```
tsob      CDRSC  CDRM=name of VTAM manager for tsob
```

If a dynamic cross-domain resource definition is not authorized, code the following CDRSC definition statement in each domain that contains a SLU that can communicate with your domain, and code a CDRSC definition statement in the domain of TSO/VTAM for each cross-domain SLU that can log on to TSO/VTAM:

```
tsobnnnn CDRSC  CDRM=name of VTAM manager for tsobnnnn
```

tsob identifies the owning CDRM for each domain and must be included on each TCAS and CDRSC definition statement. The *nnnn* suffix is a decimal integer; the numbering must start with 0001 and be sequential.

Resource Registration in an APPN Network

To prevent incorrect session setups to TSO or TSO logon failures, the default resource registration is different for TSO applications than for other applications. Applications are registered to a directory server by default because they are likely to be the object of a search. Because multiple TSO applications can be defined using the same name, registering all TSO applications would cause misleading information to be stored in the directory server. Instead TSO applications are registered according to the following conditions:

- If the APPL resource name is TSO and the ACBNAME is TSO, the default resource registration for this resource is NO registration.
- If the APPL resource name is *tsoannnn* (TSO is the prefix of the name) and the ACBNAME is *TSOnnnn*, the default registration for this APPL resource is NO registration.
- If the APPL resource name is *tsoxxxxx* and the ACBNAME is TSO, the default registration for this resource is CDSERVER (register to the central directory server).

It is assumed that AUTH=TSO is coded on each APPL definition statement.

Note: It is recommended that you not modify the registration of TSO applications from the default to avoid session setup or logon problems.

Defining TSO/VTAM Session Parameters

VTAM needs to know the session protocols required by the terminals from which individual TSO users log on. This information is provided through logon mode tables and through VTAM and NCP definition statements.

Logon mode tables associate session protocols with device types. IBM provides a default logon mode table that describes session protocols for common devices and situations. However, you might need to replace or supplement the default table.

In addition, a specific class-of-service name can be chosen. The class-of-service name used for the session is selected from the logon mode entry. If you omit the COS operand on the MODEENT macroinstruction, the default COS is chosen. To choose a specific COS, you need to include the COS name on the MODEENT macroinstruction as follows:

```
MODEENT  COS=cos name,logon mode parameters
```

The connection between a logon mode entry and a device is made using VTAM operands on the LU, LOCAL, or TERMINAL definition statements during VTAM and NCP definition. Code the DLOGMOD operand to identify the name of a logon mode table entry containing the session parameters used for the 3270 device being defined. Code the optional MODETAB operand to identify the logon mode table containing the entry. If you do not code MODETAB, the IBM-supplied logon mode table (ISTINCLM) is used.

You can also code the MODETAB and DLOGMOD operands on the GROUP, LINE, PU, LU, or CLUSTER definition statements to take advantage of the sift-down effect.

TSO/VTAM Support of 3270 Devices

TSO/VTAM supports the following two types of 3270 devices:

- Non-SNA devices attached locally or by a bisynchronous line protocol
- SNA devices attached by SDLC links.

If no logon mode table entry exists for a 3270 device, TSO/VTAM assumes that the device is non-SNA (LU0) and uses the buffer size in the SCRSIZE parameter of the TSOKEY00 member of PARMLIB. The following are valid values for the SCRSIZE parameter:

- 480 [12x40]
- 1920 [24x80]
- 2560 [32x80]
- 3440 [43x80]
- 3564 [27x132]
- 9920 [62x160]

For non-SNA devices with different screen sizes or special features and for SNA devices (LU2), you need to provide a logon mode table entry.

Use the MODETAB and MODEENT macroinstructions to define 3270 characteristics in logon mode tables and their entries. The PSERVIC operand of the MODEENT macroinstruction defines device LU type, buffer sizes, single-byte language, and QUERY capability (that is, programmed symbol sets, extended color, extended data stream, or extended highlight support). The LANG operand of the MODEENT macroinstruction defines the single-byte language of the device and whether it should be queried for double-byte capability.

Following are examples of definitions of MODEENT table entries.

MODEENT Macroinstruction for Non-SNA 3270 Devices: The FMPROF, TSPROF, PRIPROT, SECPROT, and COMPROT values shown in the following example are the same as those used in the IBM-supplied logon mode table, ISTINCLM.

```
name      MODEENT FMPROF=X'02',
          TSPROF=X'02',
          PRIPROT=X'71',
          SECPROT=X'40',
          COMPROT=X'2000',
          PSERVIC=X'...(see Table 41 on page 616)'
```

MODEENT Macroinstruction for SNA 3270 Devices: The RUSIZES operand in the following example defines a 256-byte maximum secondary logical unit RU send size and a 1024-byte maximum primary logical unit RU send size. If the maximum primary logical unit RU size is 0, TSO/VTAM defaults to 1024 bytes.

```
name      MODEENT FMPROF=X'03',
          TSPROF=X'03',
          PRIPROT=X'B1',
          SECPROT=X'90',
          COMPROT=X'3080',
          RUSIZES=X'8587',
          PSERVIC=X'...(see Table 41)'
```

PSERVIC Operand of the MODEENT Macroinstruction: Associated with each terminal is a default logon mode table entry. Within this entry, the primary and alternate screen sizes are specified, as well as a code that describes how these screen sizes are used.

The 11th byte of the MODEENT macroinstruction can have the value of X'03' (PSERVIC=X'.....03..'). When the field is specified as X'03', the primary screen size is 24x80, and the alternate screen size is determined by VTAM.

You do not have to code different logon mode table entries for TSO users when the only difference is the screen size. Users logging on to TSO with a screen size different from the default do not have to specify a logon mode table entry. The screen size is determined dynamically.

Note: The 3274 Terminal Controller must be release 65 or higher, or the logon fails.

Code the 12 bytes of device-specific hexadecimal data of the PSERVIC operand as described in Table 41.

Table 41. Coding the Device-Specific Hexadecimal Data of PSERVIC

Hexadecimal code	Related device
X'00..00000000.....00'	For non-SNA (LU0)
X'02..00000000.....00'	For SNA (LU2)
X'..00.....'	Device without extended data stream capability
X'..80.....'	Device with extended data stream capability
X'.....0000000001..'	Buffer size 480 only (12x40)
X'.....0000000002..'	Buffer size 1920 only (24x80)
X'.....0C280C507F..'	Buffer sizes 480 or 960 (12x40 or 12x80)
X'.....185020507F..'	Buffer sizes 1920 or 2560 (24x80 or 32x80)
X'.....18502B507F..'	For buffer sizes 1920 or 3440 (24x80 or 43x80)
X'.....18501B847F..'	For buffer sizes 1920 or 3564 (24x80 or 27x132)
X'.....18503EA07F..'	Buffer size 9920 only (62x160)
X'.....0000000003..'	Primary buffer size 1920 only (24x80). If a device is extended data stream capable (see above), TSO/VTAM queries the device to determine the alternate size.

To prevent switching of screen sizes on a device that has more than one size possible, code the screen size you want in the primary area of PSERVIC and a X'7E', which means no switching. Following is an example:

PSERVIC=X'.....1B8400007E..' For 3564 buffer only.

For details on coding the PSERVIC operand for your particular device, see your component description.

For details of the bit settings in the PSERVIC operand that represent bytes 13–24 of the session parameters, see “Logical Unit Presentation Services Profile” in *VTAM Programming*.

LANG Operand of the MODEENT Macroinstruction: For TSO/VTAM, this operand affects (but does not necessarily decide) the following functions:

- The language of certain TSO/VTAM terminal user messages
- The language returned on the GTTERM macro
- The single-byte character set (SBCS) filter
- The device's double byte character set (DBCS) capability.

The high order bit of LANG is used, in conjunction with other LOGMODE information, to determine if a device is queried. A device is queried if it is extended data stream capable and either the high order bit of LANG (from the MODEENT macroinstruction) is on, or the LOGMODE indicates to query for alternate screen size. The query of the device can affect each of the functions listed above.

Note: The high order bit of the LANG operand is always obtained from the logon mode table entry (specified by the LOGMODE operand). If the terminal user enters a LOGON command, the LANG or LANGTAB operands of the LOGON command will override the low order 7 bits specified by the LANG operand of the MODEENT macroinstruction.

The language of TSO/VTAM terminal operator messages IKT00201I through IKT00204I (logon failure messages)

TSO/VTAM attempts to retrieve these messages from the MVS message service (MMS), so that they can be issued in the language desired by the terminal user. The language for all other TSO/VTAM terminal user messages is set (determined) by the PLANG operand of the PROFILE command.

If the device is queried and the high order bit of LANG is on, the language obtained from the query (if known) is used for the language of the messages.

The language returned on the GTTERM macro

The TSO terminal monitor program (TMP) or command processor may issue the GTTERM macro to determine the language of a TSO/VTAM user. The PROFILE command does not affect the results of the GTTERM macro.

If the device is queried and the high order bit of LANG is on, the language obtained from the query (if known) is used for the language returned by the GTTERM macro.

The single-byte character set (SBCS) filter

If the device is queried, the appropriate SBCS filter is used (U.S. English filter, Katakana filter, or no filter). If the device is not queried, then the SBCS filter is determined from the language value.

TSO/VTAM uses this filter in TPUT data validation and translation. TSO/VTAM performs character filtering for the FULLSCR, EDIT, ASIS, and NOEDIT options of the TPUT and TGET macros.

The DBCS capability of a device

If the query reply indicates that a device is DBCS capable, TSO/VTAM enables mixed DBCS strings to be sent and received from the device. If not queried, the device is assumed incapable of DBCS processing.

If you want to disable the character filtering of TSO/VTAM, code `LANG=X'7F'` for a device that does not support double-byte capability, and code `LANG=X'FF'` for a potential double-byte character set (DBCS) device.

Note: If an incorrect LANG operand value is specified or if LANG is not specified at all, the single-byte language defaults to U.S. English.

The low order 7 bits indicate the single-byte language to be used. If a query is issued to determine double-byte capability and the high order bit of LANG is on, any supported single-byte language value returned in the query reply overrides the value specified on the LANG operand.

The following indicates the bit settings for single-byte languages:

Bit	Description
0123 4567	
0...	Indicates that the device should not be queried for language information.
1...	Indicates that the device should be queried for language information. This value should be used for devices that will potentially process DBCS data.
.nnn nnnn	Indicates the language. For a list of valid values, see "Node Initialization Block (NIB)" in <i>VTAM Programming</i> .
.111 1111	Indicates that SBCS filtering is disabled (no SBCS filter). Although not a valid language, this value can be used to disable character filtering. Note that any language other than U.S. English (X'00' or X'01') or Katakana (X'11') will also disable SBCS filtering.

Defining the 3790/3270 Configuration to TSO/VTAM

The 3270 attached to a 3790 (Version 7) uses the LU type 2 protocol. You need to give session parameters to TSO/VTAM through a logon mode table entry to properly identify the model number or screen sizes of the device being used. The 3270 operator selects FPID 932 when logging on to the 3790 to activate the 3270 data stream compatibility function. During the logon procedure, the operator is prompted for the application program ID and the logon mode table entry containing the session parameters to be used for that session. The information is transmitted to VTAM by the Initiate Self request generated by the 3790. If the operator does not supply a logon mode table entry, the 3790 uses the standard logon mode entry name EMU3790.

So that the terminal operator does not have to know what logon mode entry name to supply, create a separate logon mode table for each 3270 model attached to the 3790, and include the default entry name EMU3790 in each table with the correct session parameters for that device.

Use the MODETAB and MODEENT macroinstructions to define the logon mode table and its entry. The PSERVIC operand on the MODEENT macroinstruction carries the model number or screen sizes in row and column form. Following is an example of defining a logon mode entry for a 3270 attached to a 3790 with a screen size of 1920 bytes (Model 2).

MODETAB Macroinstruction

```
LU2TABLE MODETAB
```

MODEENT Macroinstruction

```
EMU3790 MODEENT LOGMODE=EMU3790,
                FMPROF=X'03',
                TSPROF=X'03',
                PRIPROT=X'B1',
                SECPROT=X'90',
                COMPROT=X'3080',
                PSERVIC=X'020000000000000000000200'
```

For a description of the PSERVIC values, see “PSERVIC Operand of the MODEENT Macroinstruction” on page 616.

The association between the logon mode entry and the device is made on the LU definition statement during network definition to VTAM. Use the optional MODETAB operand to identify the logon mode table containing the entry. If MODETAB is not coded, the IBM-supplied logon mode table (ISTINCLM) is used.

Defining 2741, TWX, or WTTY Terminals to TSO/VTAM

A 2741, TWX, or WTTY device attached to a communication controller can be used with TSO/VTAM through its LU type 1 protocol managers. These devices are identified to VTAM during the NCP generation process. (For details, see *Network Terminal Option Migration and Resource Definition* and the *NCP, SSP, and EP Resource Definition Guide*.) Because the NCP translates ASCII line code to EBCDIC for these devices, make sure that any logon mode entry named by the DLOGMOD and MODETAB operands on the PU or LU definition statements identifies the device to VTAM as EBCDIC. Following is an example of an ASCII TWX logon mode table entry definition that has the alternate code indicator in the COMPROT field set off (to indicate EBCDIC to TSO/VTAM).

MODETAB Macroinstruction

```
TWXTABLE MODETAB
```

MODEENT Macroinstruction

```
TWXDEVIC MODEENT LOGMODE=TWXDEVIC,
                FMPROF=X'03',
                TSPROF=X'03',
                PRIPROT=X'B1',
                SECPROT=X'90',
                COMPROT=X'3040',
                DCODE=X'00'
```

The DCODE operand on the MODEENT macroinstruction indicates to TSO/VTAM whether a TWX device is a keyboard display or a keyboard printer. If you specify DCODE to indicate that the device is a keyboard display, TSO/VTAM inhibits the display of the password when the password is entered at the device.

Defining an Interpret Table for Compatible Logons

Because TSO/VTAM uses VTAM's logon facilities, the TSO LOGON command is not supported. You can make this transparent to terminal users by defining an interpret table to allow logon requests to have the same format as that used by the TSO LOGON command. The following interpret table definition can be used:

```
INTBL    INTAB
          LOGCHAR APPLID=(APPLICID,TSO),SEQNCE='LOGON'
          LOGCHAR APPLID=(APPLICID,TSO),SEQNCE='logon'
          ENDINTAB
          END
```

This interpret table allows a user to enter a logon command in uppercase or lowercase letters.

Defining TSO to MVS

This section describes how to define TSO to MVS.

Writing a Procedure to Start TSO/VTAM Time Sharing

You need to write a cataloged procedure to start TSO/VTAM time sharing and include it in either SYS1.PROCLIB or your own procedure library. For details, refer to the *OS/VS2 MVS System Programming Library: TSO* for your operating system.

Creating a TSOKEY00 SYS1.PARMLIB Member

TSO/VTAM time sharing parameters determine how time sharing buffers should be controlled, the maximum number of users, and other options. Unless you want the IBM default parameters to apply, you must provide a SYS1.PARMLIB member containing the parameters that you want. Information on initialization and tuning for your operating system describes the parameters you can code, how to code them, and the IBM defaults.

Defining TCAS Program Properties

TCAS must have an entry in the MVS program properties table (PPT). This entry designates the execution attributes of TCAS, which should be as follows:

- The program cannot be cancelled.
- Unique protection key 6 must be assigned to the program.
- The program is privileged and is not swapped unless the address space is in a long wait.
- The program is a system task and is not timed.
- There is no host affinity.

For coding conventions, refer to initialization and tuning information for your operating system.

Implementing TSO/VTAM

This section has the following subsections:

- Translation tables
- Coding TSO/VTAM exit routines
- Security
- Performance
- 3270 large screens
- TSO considerations
- National Language Support for TSO end-user messages
- Operating VTAM under TSO
- How TSO/VTAM differs from TSO through TCAM.

Translation Tables

Translation tables allow TSO/VTAM users to replace internally those characters that are not available on a keyboard. If you call for character translation, translation tables—either your own or those supplied by IBM—are used. For more information, refer to *OS/VS2 MVS System Programming Library: TSO* for your operating system.

Coding TSO/VTAM Exit Routines

Exit routines provide the following functions needed in TSO/VTAM:

- Input and output editing that replaces or supplements IBM-supplied editing
- Attention handling that replaces IBM-supplied attention handling
- Support for terminals not supported by TSO/VTAM
- Verifying whether a terminal is capable of receiving a TSO/VTAM end-user message in the language selected by the TSO user.

For information on coding TSO/VTAM exit routines, see Chapter 4, “Writing TSO/VTAM Installation-Wide Exit Routines (MVS)” in *VTAM Customization*.

Security

The TSO subsystem is considered a secure application program. That is, confidential data is handled on behalf of the user in ways that prevent unauthorized disclosure of the data. The CONFTXT parameter in the TSOKEY00 member of PARMLIB determines whether output data is considered confidential text. The default is that the data is considered confidential. TSO/VTAM protects tracing of user data by setting the CONFTXT indicator in the NIB at the time the user logs on.

If CONFTXT=NO, VTAM can perform buffer or I/O traces on the data. If CONFTXT=YES (the default), the data is considered confidential and no data is recorded. The CONFTXT parameter, however, does not apply to the TSO type VTAM trace for TPUT/TPG/TGET buffers; these are always traceable. For details, see information on initialization and tuning for your operating system.

VTAM supports TSO message security by invoking RACF services to provide resource access control for the following:

- Cross-address space TPUTs (such as the TSO SEND command), which do the following:
 - Control who can send messages to whom

- Ensure that a message will be received by the intended user
- Ensure that a cross address space message can be received only by a user with a security classification that is equal to or greater than the sender's
- Requests to open an ACB from a non-APF authorized application program or processor.

Note: The installation must ensure that both the sender and receiver of TSO/VTAM messages are authorized with the proper security level in the security management product. The TSO/VTAM user IDs should be registered with a class of SMESAGE in the security management product.

Performance

Following are suggestions for improving TSO/VTAM performance:

- Terminal users should stack input data whenever possible. This means using the multiple-line input technique on 3270 terminals and buffered SDLC transmission on 3767 and 3770 terminals. Stacking input results in a decreased number of data transmissions to TSO. Users of 3270 terminals can stack up to a full screen of data; 3767 and 3770 terminal users can stack from 256 bytes to 2000 bytes of data, depending on the terminal's buffer size.
- Users of 3767 and 3770 terminals can reduce idle time by "typing ahead." This means that whenever the terminal is not receiving data or is transmitting data, the user can enter input.
- Users of 3276 and 3278 terminals can expedite unlocking the keyboard if you use BREAK mode. BREAK mode reduces idle time by allowing typing ahead.
- Users of 3270 terminals can shorten the TSO/VTAM processing path when handling input if you use automatic line numbering. This reduces system overhead and expedites unlocking the keyboard.

3270 Large Screen Considerations

The following information describes screen management techniques for IBM 3270 terminals that have default and alternate screen sizes determined by the logon mode table entries used for them at logon. The default size is accessed by sending an Erase/Write (EW) command to the terminal, and the alternate size is accessed by sending an Erase/Write Alternate (EWA) command.

TSO/VTAM Screen Management

TSO/VTAM can manage the screen for TSO application programs that use line-oriented I/O. In this case, TSO/VTAM uses the maximum screen size for the device, whether it is the default or the alternate in the logon mode table entry. When screen erasure is necessary, TSO/VTAM uses either the EW or the EWA command as required to access the larger screen size.

Full-Screen Application Program Screen Management

An IBM-supplied or user-supplied application program that uses full-screen I/O in TSO full-screen mode can be used to manage the screen. TSO/VTAM uses the screen size used by the application program. If the CLEAR key is pressed, TSO/VTAM erases the screen with either an Erase/Write (EW) or Erase/Write Alternate (EWA) command depending on which was last used by the application program.

TSO Considerations

Information about the TSO-related steps that must be performed before a TSO/VTAM terminal user logs on is in *OS/VS2 MVS System Programming Library: TSO*, and in information on initialization and tuning for your operating system. (Make sure you have updated these books with TSO/VTAM supplements before referring to them.) The following list outlines the steps and refers you to the appropriate publication:

- Write LOGON cataloged procedures, and include them in SYS1.PROCLIB. Refer to *OS/VS2 MVS System Programming Library: TSO* for your operating system.
- Construct the TSOKEY00 member or an alternate member of SYS1.PARMLIB (or an alternate data set) to set VTIOC parameters. Refer to information on initialization and tuning for your operating system.
- Include SYS1.CMDLIB in a LNKSTxx member of SYS1.PARMLIB or in a LOGON cataloged procedure. Refer to information on initialization and tuning for your operating system.
- Create or convert the user attribute data set (UADS) and the broadcast data set. Refer to *OS/VS2 MVS System Programming Library: TSO* for your operating system.
- Build translation tables if you want character translation. Refer to *OS/VS2 MVS System Programming Library: TSO* for your operating system.
- Write the procedure that starts TSO/VTAM time sharing. Refer to *OS/VS2 MVS System Programming Library: TSO* for your operating system.
- Write any command exit routines you plan to use. Refer to *OS/VS2 MVS System Programming Library: TSO* for your operating system.

Note: You are strongly encouraged not to VARY LOGON to TSO or code TSO as a LOGAPPL because unpredictable results can occur. For instance, if the original BIND fails, TSO messages and VTAM messages may be repeatedly issued to the console indicating the session setup failure. In some cases this may not stop even if the VARY NOLOGON command is used to break the automatic logon relationship.

National Language Support for TSO/VTAM End-User Messages

You can tailor TSO/VTAM end-user messages using the MVS Message Service (MMS). You define TSO/VTAM end-user messages to MMS the same way you define USS end-user messages. For details, see “Defining USS Messages to the MVS Message Service” on page 317.

The TSO user selects the language used for TSO/VTAM messages using the PROFILE command with the PLANG operand. TSO/VTAM then retrieves its messages from MMS using the language specified by the user. This language continues to be used until the TSO/user changes the language (by reissuing the PROFILE command).

Before the TSO user can issue the PROFILE command, TCAS may issue messages IKT00201I through IKT00204I. The language used for these messages is specified on either the LANG operand of the MODEENT macroinstruction or on the USS logon command using the LANG or LANGTAB operands. The language is passed from these sources to the TSO/VTAM logon exit.

For information on how MMS translates messages, see Appendix G, “Message Translation Using the MVS Message Service” on page 681.

Operating VTAM under TSO

A TSO operator is defined as any TSO user with the authority to use the CONSOLE command. MVS or subsystem operators running under TSO are treated as extended multiple consoles. Authorized users can establish a console session with MVS console services. After the session is active, users can enter MVS and subsystem commands and obtain solicited and unsolicited messages.

By representing all consoles and operators as a 4-byte console ID, multiple console support (MCS) can extend its current command and message services to TSO operators. This allows TSO operators to send VTAM commands and receive VTAM messages.

To enable this function, you need to grant individual users CONSOLE command authority, so that they can use the MVS CONSOLE command. There are three ways that you can give CONSOLE authority to users:

- Using RACF facilities (RACF RDEFINE and RACF PERMIT commands)
- Using the TSO logon pre-prompt exit routine
- Using the TSO CONSOLE exit routine.

For information on giving CONSOLE authority, the levels of authority, and the CONSOLE command, see *TSO Extensions Version 2 Customization*.

A TSO operator can use the MVS Message Service (MMS) to retrieve translated messages, including VTAM, logon manager, and TSO/VTAM messages for network operators. The TSO operator can select a language using the PROFILE command with the PLANG operand. For information on how MMS translates messages, see Appendix G, “Message Translation Using the MVS Message Service” on page 681. For details on defining translated network operator messages, see “Unformatted System Services (USS) Tables” in the *VTAM Resource Definition Reference*.

How TSO/VTAM Differs from TSO through TCAM

The following list describes differences between TSO/VTAM and TSO through TCAM. It is intended for system programmers who are familiar with TSO through TCAM.

- TSO/VTAM does not use the TCAM message control program (MCP).
- VTIOC replaces TIOC (terminal I/O coordinator) as the means for controlling data movement between TSO and the access method.
- TSOKEY00 replaces IKJPRM00 as the SYS1.PARMLIB member that contains time-sharing parameters.
- The system START command starts TSO/VTAM time sharing. The MODIFY command controls the number of users that can be logged on and terminates user address space. The STOP command stops TSO/VTAM time sharing.
- TSO/VTAM user address spaces are totally independent of each other. Each address space is an individual TSO/VTAM application program with ownership of its terminal. A discontinuous terminal status block (TSB) is allocated dynam-

ically when each user logs on. Users do not share buffers; VTIOC allocates buffers as needed.

- TSO/VTAM uses VTAM's logon facilities. The TSO LOGON command, which provides a logon to the TSO terminal monitor program (TMP), is supported only for re-logons. The TSO LOGOFF command is supported only if HOLD is not coded.
- TSO/VTAM support for the PROFILE and TERMINAL commands differs from that provided through TCAM. You can define character-delete and line-delete control characters (PROFILE command) only for the LU type 1 devices supported by TSO/VTAM because of editing capabilities of TSO/VTAM-supported terminals. Definition of attention interrupt characteristics (TERMINAL command), including the simulated attention interrupt, is not supported because attention interrupts are always accepted on TSO/VTAM-supported terminals. Character translation (TERMINAL command) is provided for TSO/VTAM-supported terminals.
- There are some considerations for terminal control macroinstructions. STTRAN sets character translation. STFSMODE turns 3270 full-screen mode on or off. STLINENO is the number of a 3270 screen line on which the next non-full-screen message should appear and optionally turns full-screen mode on or off. STTMPMD determines whether the terminal control routine is active for the terminal and whether the ATTN and CLEAR keys are to be passed to the application program as data. GTTERM obtains the primary or alternate screen sizes and places them in a user-provided area. The STATTN macroinstruction is not supported by TSO/VTAM. The STCC macroinstruction is supported for NTO-attached devices.
- No data is lost when a user logs on again (reconnects) after his line is disconnected.
- TSO/VTAM takes advantage of multiprocessing because it is multitask structured; TCAM runs in one host processor at a time.

Appendix B. VM/SNA Console Support

VM/SNA console support (VSCS) is a VTAM application program that allows a terminal in an SNA network to act as the console of a virtual machine. VSCS acts as an intermediary between VTAM and the VM operating system. If end users in a VTAM network want to log on to VM, they must first log on to the VSCS application program running in the VM host. VSCS, however, can be made transparent to end users.

Refer to *Planning for NetView, NCP, and VTAM* for a list of the devices supported by VSCS as virtual machine consoles and for a list of the output devices supported by VSCS.

Preparing to Start VSCS

When you install VTAM (as described in Chapter 3, "Installing VTAM under VM" on page 51), you take steps to allow VSCS to run in the same virtual machine as VTAM. If you need to run more than one VSCS in the same GCS group, the second VSCS must run in another virtual machine.

This section describes the following steps that you must perform before starting VSCS in a virtual machine:

1. Define VSCS as a VTAM application program.
2. Authorize the VSCS virtual machine.
3. Update the directory entry for the VSCS virtual machine.
4. Prepare the GCS environment to start VSCS.

Defining VSCS as a VTAM Application Program

You must define VSCS to VTAM in an application program major node. Refer to "Defining Application Programs" on page 138 for general information about defining application programs.

ISTAPPLS VTAMLST is the name of the file containing a sample definition statement for the VSCS application. This sample is only part of the definition of an application program major node. You need to use the sample VSCS APPL definition statement in the definition of your application program major node.

The following example shows the definition statement required to define the VSCS application program to VTAM:

```
ISTAPPLS VBUILD TYPE=APPL
VM      APPL  AUTH=(PASS,ACQ),ACBNAME=VM,PRTCT=VM,AUTHEXIT=YES,
        SONSCIP=YES
```

AUTH=(PASS,ACQ) authorizes VSCS to use needed VTAM functions.

ACBNAME=VM specifies the name of the minor node assigned to the VSCS application program. The name specified must match the name used on the APPLID operand of all DTIGEN macroinstructions that are coded for the VSCS associated with this APPL definition statement. If you do not code an ACPNAME, the default name is VM.

PRTCT=VM specifies a password used to verify that an application program is authorized to run (as the application program being defined by this APPL definition statement). If you code the PRTCT operand, the password you specify must match the password coded on the PASSWRD operand of all DTIGEN macroinstructions that are coded for the VSCS associated with this APPL definition statement. If you do not code the PRTCT operand, VTAM does no password checking.

AUTHEXIT=YES is required. This operand allows VSCS's VTAM exit routines to run in supervisor state.

SONSCIP=YES allows VTAM to terminate a session on behalf of VSCS (without issuing a CLSDST macroinstruction). When this happens, VTAM schedules VSCS's SCIP exit routine with an UNBIND RU that contains information (type codes) describing the nature of the failure. This operand is required for VSCS recovery to function properly.

Note: If you use either the VARY LOGON operator command or the LOGAPPL operand on the LU definition statement to connect LUs to VSCS, VTAM automatically attempts recovery of the devices. Therefore, in these two cases, you should not code the SONSCHIP operand on the APPL definition statement for VSCS. Not coding SONSCHIP prevents VTAM and VSCS from attempting recovery of the same devices.

Do not code any operands on the APPL definition statement other than those shown in the example.

Note: If you are running more than one VSCS, each one must have its own VTAM APPL definition statement with a unique ACBNAME specified. This name must match the name specified on the APPLID operand of the DTIGEN macroinstruction. Only one VSCS can be active for each virtual machine.

Authorizing the VSCS Virtual Machine

If VSCS runs in a virtual machine other than the VTAM virtual machine, that VSCS virtual machine must be authorized in the GCS virtual machine group. Refer to *VM/ESA Group Control System* for instructions on how to authorize a virtual machine.

Updating the Directory Entry for the VSCS Virtual Machine

To provide SNA console support, VSCS uses the following:

- Console communications services (CCS), a facility of CP that provides full VM console interface capabilities for SNA terminal users
- The inter-user communication vehicle (IUCV), which transfers messages between VSCS and CCS.

The VM directory entry of the VSCS virtual machine must contain the following:

- An IUCV control statement to allow VSCS to communicate with CP
- An OPTION control statement with MAXCONN to specify how many SNA terminals are supported at one time.

Following is an example of the directory entry for VSCS:


```

USER VSCS VSCS 10M 64M ABG
OPTION DIAG98 MAXCONN 400
IUCV *CCS P M 10
IUCV ANY P M 0
ACCOUNT VTAM GCS
IPL GCS PARM AUTOLOG
CONSOLE 01F 3215
SPOOL 00C 2540 READER A
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK MAINT 190 190 RR
LINK MAINT 595 595 RR
LINK MAINT 298 191 WR
LINK MAINT 29A 29A RR

```

Notes:

1. The VSCS virtual machine directory entry must specify the size of the VSCS virtual machine.
2. The DIAG98 option is strongly recommended for the VSCS user ID.

See the *VM/ESA Planning Guide and Reference* for more information on VM directory entries. Also, refer to Chapter 3, "Installing VTAM under VM" on page 51 for an example of a VM directory entry coded for the VTAM virtual machine.

If you are using real I/O operations in the VTAM virtual machine (specified by DIAG98), then you should also specify DIAG98 on the option control statement in the VSCS virtual machine. This enables the VSCS virtual machine to page lock necessary buffers.

Preparing the GCS Environment to Start VSCS

Before the VSCS START command is issued to start VSCS, the operating system environment (GCS) must be established.

Do the following to prepare the GCS environment so that VSCS can be started. These steps assume that GCS is running in the VSCS virtual machine and that the necessary setup, described in the preceding sections, has been done.

1. Access the RUN minidisk, if it is not already accessed. This disk contains the VSCS load library.
2. Access the minidisk that contains the VSCSUSER load library. If you are not using the defaults to start VSCS, substitute the name of your load library if it is other than VSCSUSER.
3. Issue the GCS GLOBAL LOADLIB command to specify the names of the VSCS and VSCSUSER load libraries. Following is an example of the GLOBAL command:

```
GLOBAL LOADLIB VSCSUSER VTAM VSCS NCP
```

4. Issue the GCS LOADCMD command to load the VSCS operator command processor (DTISLCMD), and define the VSCS operator command prefix (usually VSCS). All VSCS operator commands must be preceded by a prefix, which is determined by the first operand on the GCS LOADCMD. Following is an example:

```
LOADCMD VSCS DTISLCMD
```

5. Issue the following command:

```
CP ENABLE SNA
```

6. Issue the following START command to start VSCS:

```
VSCS START [PARM=n]
```

n is a single digit in the range 0–9 that corresponds to a DTIUSER*n* load module in the VSCSUSER LOADLIB. If you leave out PARM=*n*, VSCS is started with a set of default initialization parameters and options.

You can issue the VSCS START command from the following:

- The console of the VTAM and VSCS virtual machine
- The PROFILE GCS for VTAM and VSCS.

If you use the VM autolog facility in conjunction with the PROFILE GCS, you can start VSCS without operator intervention when an IPL is done for the VM system.

Normally, the VSCS START command completes with a 0 return code after VSCS is ready to accept additional commands. At this point, you can issue VSCS and GCS commands from the console of the VTAM and VSCS virtual machine while initialization and execution are still in progress. If VSCS initialization fails before reaching this point, you get a nonzero return code.

You can start VSCS independently of VTAM or even before starting VTAM. VSCS initialization periodically retries startup until VTAM is initialized and the VSCS application program name is activated. Retries occur every 30 seconds and continue for the number of minutes indicated by the ACBLOOP VSCS initialization parameter.

Tailoring VSCS

This section describes the use of the following:

- The VSCS CHANGE command to change start options
- Logon mode tables to tailor session parameters for devices supported by VSCS
- USS tables to tailor LOGON and LOGOFF commands
- VSCS data manipulation exit routines to tailor logical unit input and output data streams
- Protecting confidential data.

Changing VSCS Start Options

The VSCS CHANGE command allows you to dynamically change most of the VSCS start options. This command enables you to tune VSCS without having to take the system down and bring it back up again. However, the changed start options last only while VSCS is active. To make them permanent, change the VSCS start options using the DTIGEN macroinstruction. Alterations made with the CHANGE command could lead to VSCS termination, or reduced performance. Therefore, use this command with caution.

The VSCS CHANGE command also gives you control over the VSCS dump process. Several of the changeable VSCS start options enable you to set the number of internal repetitive and system repetitive dump requests that are allowed within a specified time.

Note: Use the VSCS STAT TRACE command to record storage utilization and changes resulting from the use of the CHANGE command.

Session Parameters for Devices: Logon Mode Tables

Logon mode entries from an MVS/VTAM host do not provide the necessary logon information to a VM/VTAM host when logging on to VSCS (unless the device type is a 3277). If an LU that is owned by a non-VM/VTAM host attempts to log on to VSCS (in a VM/VTAM host), the default logon mode entries might not provide the necessary logon information. This will result in errors for the LU in session with VSCS.

To provide support for devices not supported by the IBM-supplied logon mode tables, you will need to code your own MODEENT macroinstructions for SNA and non-SNA 3270 devices. See *VTAM Resource Definition Reference* for information on coding the MODEENT macroinstruction.

VSCS supports WSF QUERY to dynamically determine screen sizes for 3290 displays.

IBM supplies several logon mode tables. ISTINCLM, the default table, contains logon mode entries for use with VSCS and the NetView program. ISTINCLM contains defaults for most devices you are using. ISTTABLE contains entries for terminals that might not be using VSCS (for example, terminals that need to log on to VTAM application programs running under MVS, such as TSO).

In addition to these two tables, IBM supplies eight tables for use with the NetView program and the terminal access facility (TAF). Refer to the block comments provided on each entry under the descriptions of IBM supplied tables. See the default logon mode tables for VM in the *VTAM Resource Definition Reference* for an example of the following tables:

```
DSIASCII
DSIASCPR
DSICNMDT
DSILU0
DSIXDOM
DSI4LU2
DSI6LU2
TAFBINDS.
```

You specify a logon mode by specifying the following names on a TERMINAL, LOCAL, or LU definition statement that defines an LU:

- On the MODETAB operand, the name of the logon mode table
- On the DLOGMOD operand, the name of the logon mode entry within the logon mode table.

If MODETAB and DLOGMOD are omitted from the logical unit's definition statement, they can be obtained by the sift-down effect. (The *VTAM Resource Definition Reference* identifies and describes the definition statements and operands to

which sifting applies.) If MODETAB and DLOGMOD are not coded anywhere, MODETAB defaults to ISTINCLM, and DLOGMOD defaults to the first entry in the logon mode table. A terminal user who is logging on to VSCS can override the logon mode entry (DLOGMOD) for his or her LU by entering the following statement, where name is the name of the logon mode entry that overrides DLOGMOD:

```
LOGON(VM) LOGMODE(name)
```

Specification of an incorrect logon mode or an error in the logon mode entry itself can result in errors for an LU in session with VSCS, including the following:

- Loss of color
- Incorrect screen format
- Incorrect characters
- Logon problems
- Logical unit being purged.

Before creating a new logon mode entry, check to see if an entry already exists in ISTINCLM for that device. See the *VTAM Resource Definition Reference* for the IBM-supplied logon mode tables and for information on how to create or modify VSCS logon mode entries.

LOGON and LOGOFF Commands: USS Tables

USS LOGON and LOGOFF commands to the needs of your terminal operators. When you first install VTAM and VSCS, there is a default USS table that defines the USS LOGON and LOGOFF commands. This table also defines VM to result in LOGON APPLID(VM).

See the *VTAM Resource Definition Reference* for the default USS table and for information on coding supplemental USS tables.

Data Streams: VSCS Data Manipulation Exit Routines

VSCS data manipulation exit routines allow for customization of logical unit input and output data streams. These routines can modify a data stream before VSCS translates it and can prevent VSCS from translating the data stream altogether.

VSCS data manipulation exit routines allow data stream translation for keyboard/printers (2741, 3767), TWX devices (3101), and display devices (such as the 3270). They allow for translation of the following data:

- Data sent to a display device
- Data received from a display device
- Data sent to a keyboard/printer or TWX device
- Data received from a keyboard/printer or TWX device.

The VSCS options DEXIT and KEXIT must be coded as follows:

- DEXIT=Y to activate the exit routines for display devices
- KEXIT=Y to activate the exit routines for keyboard/printers and TWX devices.

For information on coding VSCS exit routines, see *VTAM Customization*.

Protecting Confidential Data

When VSCS is transmitting or receiving confidential data, you can clear the data from VTAM buffers for added security. To take advantage of this security option, code `CONFTXT=Y` on the DTIGEN macroinstruction. For information on coding the DTIGEN macroinstruction, see the *VTAM Resource Definition Reference*. Otherwise, you can use the `CHANGE` command; see *VTAM Operation*.

VSCS Start Options

VSCS start options are specified as operands on the DTIGEN macroinstruction. These operands include both VSCS initialization parameters and VSCS options. The VSCS options can be updated after VSCS is initialized using the VSCS `CHANGE` command.

The following sections explain how to choose values for the VSCS initialization parameters and options. For information on coding the DTIGEN macroinstruction, refer to the *VTAM Resource Definition Reference*.

Controlling VSCS Storage Management

The VSCS initialization parameter `BLKMULT` is a multiplier that is used during VSCS initialization to increase the storage segment size. For example, if `BLKMULT` is set to 2, VSCS obtains an 8K storage segment instead of 4K, or 16K instead of 8K. This not only reduces the number of GETMAINS issued, but also reduces the storage manager's capability to swap or release available storage segments.

Two VSCS options, `STCHKTM` and `STRELTm`, allow some control over VSCS dynamic storage. These VSCS options can be specified either before initialization as operands on the DTIGEN macroinstruction or after initialization by using the VSCS `CHANGE` command.

Storage segments, as discussed here, have the following two forms:

- An unavailable storage segment—one or more blocks of the segment are currently being used (segment is allocated and in use).
- An available storage segment—no blocks of the segment are currently being used (segment is allocated but not in use).

Note: When a GCS GETMAIN fails to obtain private storage, VSCS detects a storage shortage and returns all available storage segments to GCS. This function is independent of the storage checking and releasing that are controlled by `STCHKTM` and `STRELTm`.

When the `STCHKTM` time interval has expired and VSCS has no additional LU work to do, the following occurs:

1. VSCS scans each of the VSCS storage pools for available storage segments.
2. If more than one segment of available storage is found in a pool, the following occurs:

- a. If the storage segment is larger than 12K² in size, a FREEMAIN is issued to return the storage to GCS.
- b. Half of the available storage segments are removed from this storage pool and made accessible for use by all VSCS storage pools (not returned to GCS).
3. If only one segment of available storage is found, that segment is neither removed from the storage pool nor returned to GCS.
4. If exception response mode is being used, each LU is checked for extra storage used for recovery. If the extra storage is found and the extra storage has been queried for at least sixty seconds, it will be freed up.
5. If exception response mode is being used, each LU with an active RPL will be checked to see if the VTAM request has been active for more than ninety seconds. If so, then message DTIS57I will be issued.

When the STRELTm time interval has expired and VSCS has no additional LU work to do, VSCS returns to GCS half of the storage segments that are currently accessible for use by all storage pools. (These are the storage segments that were removed from storage pools in step 2b.) A FREEMAIN is issued for each of these storage segments.

The following table shows the results of different settings of the STCHKTM and STRELTm options.

STCHKTM value	STRELTm value	Result
0	0	No storage checking or releasing is done (the default).
0	R (R>0)	No storage checking or releasing is done. Do not use this combination of settings; instead, for the same results, use the default settings. If STCHKTM=0, storage checking is not done, and therefore, storage releasing (freeing excess storage found during storage checking) is futile.
C (C>0)	0	Storage checking is done every C seconds: half of available storage segments are made available for all storage pools.
C (C>0)	R (R>0)	Storage checking is done every C seconds: half of available storage segments are made available for all storage pools. Also, storage releasing is done every R seconds: half of the storage segments made available for all storage pools (by storage checking) are returned to GCS using the FREEMAIN macro.

Note: When you choose values for STCHKTM and STRELTm, consider the following:

- After a single storage check and a single storage release, approximately one-fourth of the available segments are returned to GCS.
- After the next check and release, approximately one-fourth of the remaining available segments are returned to GCS.

² This value is multiplied by the VSCS option BLKMULT at VSCS initialization time. The default value of BLKMULT is 1.

This method prevents VSCS from issuing a large number of FREEMAINs (during periods when there are temporarily a large number of available storage segments) followed by a large number of GETMAINs (to obtain storage that has just been released). Also, over time, this method reduces the amount of storage that is not being used.

Following are recommendations for use of the STCHKTM and STRELTm options:

- If storage is not limited, use the STCHKTM option or both the STCHKTM and STRELTm options.
- If storage is limited, use both of these options, setting STRELTm greater than STCHKTM.
- If machine cycles are limited and storage is not limited, use the STCHKTM option or the STCHKTM and STRELTm options with large time values specified; or set both options to 0 (default).
- If machine cycles are limited and storage is limited, use either the STCHKTM option or the STCHKTM and STRELTm options.
- If you are unsure of the conditions mentioned in the preceding steps (flexible load), use the STCHKTM option.
- If you are migrating from VCNA, initially use the defaults.
- If you are a new VSCS user, initially use the defaults.

Specifying STRELTm < STCHKTM reduces dynamic storage and increases the number of GETMAINs and FREEMAINs issued. For normal environments, you should specify STCHKTM < STRELTm, or STCHKTM alone because this reduces the number of GETMAINs used. The time values chosen should not be too low because of additional processing involved in scanning the VSCS storage pools, but they should not be so high as to reduce the effectiveness of the feature.

Controlling VSCS Dumps

The VSCS options DMPFREQ, DMPSYSC, and DMPINTC allow for control of the number of dumps taken by VSCS. These VSCS options can be specified either before initialization as operands on the DTIGEN macroinstruction or after initialization by using the VSCS CHANGE command.

When a VSCS dump request is received, the VSCS dump processor evaluates the request before actually taking the dump. The following VSCS options are used to decide if a dump should be taken:

- DMPINTC is the number of repetitive internal dumps³ that are allowed during the DMPFREQ time interval. The default value, 0, means there is no limit to the number of repetitive internal dumps allowed.
- DMPSYSC is the number of repetitive system dumps⁴ that are allowed during the DMPFREQ time interval. The default value, 0, means there is no limit to the number of repetitive system dumps allowed.

³ Internal dumps are dumps requested by VSCS when an error condition is detected. Repetitive internal dumps are dumps that are requested for the same error condition.

⁴ System dumps are dumps requested by the VSCS ESTAE routine, which receives control after a VSCS task ABENDs. System dumps are repetitive if they have the same ABEND completion code; however, rarer ABEND completion codes are grouped together so that system dumps with these codes are considered repetitive.

- **DMPFREQ** specifies the time interval (in seconds) in which only DMPINTC repetitive internal dumps and DMPSYSC repetitive system dumps are allowed. The default value, 0, means the time interval is set to the entire time that VSCS is active; thus, after the DMPINTC and DMPSYSC limits are reached, no more dumps are taken by VSCS.
- **DMPTIME** specifies the number of seconds that the GCS SDUMP macroinstruction process takes to complete. DMPTIME is not directly used to decide if a dump should be taken. Instead, this value, which defaults to 90 seconds, is only used to validate the DMPFREQ value to make sure that it is not set too low.

Note: If the value of DMPFREQ is less than the number of seconds required to dump virtual storage, VSCS allows an unlimited number of dumps. This happens because by the time the first dump completes, the next time period (of DMPFREQ seconds) is entered; therefore, the next request for a dump is accepted regardless of the settings for DMPSYSC or DMPINTC.

The following table shows the results of different settings of the DMPFREQ, DMPSYSC, and DMPINTC options.

DMPFREQ value	DMPSYSC or DMPINTC value	Result
0	0	No limits are placed on the number of dumps taken by VSCS. This is the default.
0	N (N>0)	During the entire time that VSCS is active, a maximum of N repetitive dumps are allowed for each unique type of error.
T (T>0)	0	No limits are placed on the number of dumps. This functions the same as the default setting (DMPFREQ is treated as if it were 0).
T (T>0)	N (N>0)	During a time period of T seconds, a maximum of N repetitive dumps are allowed for each unique type of error.

Controlling VSCS Performance

VSCS performance is affected by the following options and initialization parameters.

Before VSCS is initialized, you can specify these options and parameters as operands on the DTIGEN macroinstruction. After VSCS is initialized, you can use the VSCS CHANGE command to update them except for RPLNUM, SCHED, and SPEC.

AQLIMIT

specifies the maximum amount of storage for exception response mode PIUs (PIUs that do not require a definite response) that are queued for each LU. The larger the value, the more exception response PIUs VSCS generates, but also the more private storage VSCS uses. This option is valid only when running with the SCHED=Y option (exception response mode).

DPACE

determines the number of line-by-line writes to a display LU that can occur before a response is required. The normal default is 0, which sets the value to the depth of each display screen and allows a whole screen of data to be sent

before requiring a response. If the value coded is too small, additional I/O might be required.

DPXMTL

determines the size of the output buffer used for display LUs when transmitting line-by-line rather than full screen data. The default value should be sufficient for most environments. The use of a large value requires additional storage. If storage is limited, use a smaller value.

FSREAD

can prevent the loss of data by retrieving read buffer data from an LU before switching out of full screen mode. If the VSCS option FSREAD is not used, the extra I/O for the read buffer does not occur, but data can be lost for application programs that retrieve input data with a read buffer.

KPACE

determines the number of line-by-line writes to a keyboard/printer LU before a response is required. The default should be sufficient for most environments, but can be lowered if you want a faster response to the Attention/Break key.

Note: KPACE does not apply to TWX devices or devices that have a pace value of 1.

KPXMTL

determines the size of the output buffer used for keyboard/printers and TWX LUs. The default value should be sufficient for most environments. However, if you are using larger line sizes (as with the CP TERMINAL command), this value should be increased to prevent problems with data truncation.

QDEPTH

determines the number of pending requests allowed for an LU. When this value is exceeded, the LU is purged. The default and maximum value is 32600, and the minimum value is 200.

RCVBFRL

determines the size of the RECEIVE buffer used for RECEIVE SPECIFIC or RECEIVE ANY when receiving input data from an LU. Setting this option to a larger value prevents the necessity for most RECEIVE SPECIFIC requests (when VSCS is using the RECEIVE ANY method of obtaining LU input), but increases the VSCS storage requirements.

RPLNUM

determines the number of RECEIVE ANY RPLs used (when the VSCS option SPEC is not used). Increasing the value of this option increases the storage requirements slightly, but benefits performance when the number of VSCS LUs exceeds 40.

SPEC

used for all devices. It causes VSCS to issue a VTAM RECEIVE SPECIFIC for each LU. This means that a VTAM RESETR is no longer needed to switch an LU from SPECIFIC to ANY mode. RECEIVE SPECIFIC requires additional storage because each LU has its own RECEIVE RPL and RECEIVE buffer. (The size of the RECEIVE buffer is controlled by the VSCS option RCVBFRL.) RECEIVE SPECIFIC causes VTAM to require additional CRPLs; you should consider this when you set up VTAM storage pools.

SCHED

Affects all devices except printers. Enables the use of exception response VTAM SENDs when sending output requests to a logical unit. This reduces the number of definite response VTAM SENDs issued, thus reducing network traffic and definite response processing. This option requires additional storage to provide a limited error recovery process. Error situations that are recoverable by the user's VM application are limited, and an increase in the number of users being disconnected can occur.

VSAMLM

sets the IUCV message limit allowed for each LU. This determines how many outstanding messages are allowed for each LU before additional requests are rejected. In addition, this option is used to determine the number of virtual external interrupt buffers that VSCS allocates for each LU. Specifying a large value for this option can have adverse effects on performance.

Controlling VSCS SCIP Recovery

The following VSCS options control the SCIP recovery process.

- Number of recovery attempts allowed for each LU:
 - SCIPCNT
- Saturation prevention:
 - SCIPNUM
 - SCRIPTIM.

These options can be specified either before initialization as operands on the DTIGEN macroinstruction or after initialization by using the VSCS CHANGE command.

The following table shows the results of different settings of the SCIPNUM and SCRIPTIM options.

SCIPNUM value	SCRIPTIM value	Result
0	0	Allows an unlimited number of LUs to enter SCIP recovery (the default).
0	T (T>0)	Allows an unlimited number of LUs to enter SCIP recovery. The value of SCRIPTIM is treated as if it were 0.
N (N>0)	0	Allows an unlimited number of LUs to enter SCIP recovery. The value of SCIPNUM is treated as if it were 0.
N (N>0)	T (T>0)	Limits N LUs to enter SCIP recovery, during each T second time interval.

By using SCIPNUM and SCRIPTIM, you can limit the number of LUs (SCIPNUM) that are allowed to begin recovery during a specified time interval (SCRIPTIM). This prevents VSCS from attempting recovery of all failing LUs at one time (which can have a major impact on the performance of the LUs that are still active). Also, if you limit the number of LUs that are allowed to begin recovery, the rate of successful recovery attempts increases.

Note: If you use either the VARY LOGON operator command or the LOGAPPL operand on the LU definition statement to connect LUs to VSCS, VTAM automatically attempts recovery of the devices. Therefore, in these two cases, you should not code the SONSCIP operand on the APPL definition statement for VSCS. Not

coding SONSCIP prevents VTAM and VSCS from attempting recovery of the same devices.

Defining Your Own Set of VSCS Start Options

You should code your own set of VSCS initialization parameters and options if the default values do not meet your needs. You can code up to ten different sets of values for VSCS. Each set must be in a unique DTIUSER n ASSEMBLE file; each file must contain exactly one DTIGEN macroinstruction; and each file must be assembled separately using the VTAMBLD MACLIB. The resulting text files are link-edited into a user loadlib, that is, VSCSUSER LOADLIB on the LOCALSAM minidisk. DTIUSER n files are coded with the IGNORE option in the VSCSUSER LOADLIB, ISTBLVL8. Following are the steps required to link-edit your text files into the VSCSUSER LOADLIB:

1. Edit ISTBLVL8 and remove the IGNORE option for the appropriate DTIUSER n file.
2. Copy the appropriate DTIUSER n ASSEMBLE file onto your A-disk and modify it as necessary.
3. Issue the global command as follows:

```
GLOBAL MACLIB VTAMBLD
```

4. Assemble your DTIUSER n file as follows:

```
VMFHASM DTIUSER $n$  5654010A VTAM (PPF
```

This creates a DTIUSER n TEXT file on your A-disk.

5. Copy the DTIUSER n TEXT file from the A-disk to the 2C4 disk (accessed as the E disk) as DTIUSER n TXTnnnnn, where nnnnn is a user-defined number usually starting with 00001.
6. Erase the DTIUSER n TEXT file from your a disk.
7. Relink DTIUSER n into ISTBLVL8 using the following command:

```
VMFBLD PPF 5654010A VTAM ISTBLVL8 DTIUSER $n$  (ALL
```

For more information on making local modifications, see "Installing Tables and Modules in VM" on page 86.

Refer to the *VTAM Resource Definition Reference* for information on how to code the DTIGEN macroinstruction.

After you have created a set (or sets) of VSCS start options, start VSCS with the VSCS START command. Unlike VTAM, VSCS does not prompt for options and does not allow you to enter alternate options on the command line. The format of this command is as follows, where n is a single digit in the range 0–9 that corresponds to a DTIUSER n load module in VSCSUSER LOADLIB:

```
VSCS START [PARM= $n$ ]
```

If you omit PARM= n , VSCS is started with the IBM-supplied set of default start options.

Using VSCS Diagnostic Tools

The VSCS component of VTAM for VM offers its own set of diagnostic tools. There are commands to control tracing, dumping, and buffer allocation. Also, one command lets you query the status of diagnostic tool options.

All VSCS commands can be abbreviated, except the START, HALT, QUIT, and CANCEL commands.

Tracing Commands

VSCS provides a trace facility that can record the following:

- Data passed over the inter-user communication vehicle (IUCV) interface. (VSCS makes entries in the trace table for all requests that flow over the IUCV interface in both directions.) The data is limited to the first 4 bytes.
- VSCS dispatcher activity.
- Invocation of the VTAM API exits.
- Clean-up activity.
- User data to and from the LU.
- Get block and free block.
- Storage statistics (external only)
- Operator commands and messages

Tracing can be limited to activity involving selected logical units, and can be recorded externally.

The VSCS TRACEON command turns on VSCS traces. The options DISP, CCS, VTAM, GET, FRE, DAT, STAT, OPER, ILU, ALL, and EXT control which traces are turned on and whether the trace records are externally recorded (trace option STAT must be turned on separately, and records are externally recorded only when STCHKTM or STRELTM is greater than 0). The VSCS CCS and VTAM traces default to active when VSCS is started. VTAM dispatcher entries are traced, by default, when VSCS is activated. The trace table holds 2000 entries by default. However, you can change the size of the trace table by using the DTIGEN operand TRASIZE. See the *VTAM Resource Definition Reference* for more information on the DTIGEN macroinstruction.

For detailed trace data information, refer to *VTAM Diagnosis*.

The VSCS TRACEOFF command turns off VSCS traces. When the trace is turned off, the storage is not freed. No further entries, however, are made until the trace is turned back on. Effectively, the table is frozen and can be dumped by the operator using the VSCS DUMPLU command, or using a GCS or CP command to dump the entire virtual machine. To accommodate this technique, the address of the trace table (its beginning and its end) is provided when the table is allocated. It is also provided in messages generated by the VSCS QUERY command.

Tracing helps to pinpoint the failing component of a network. You can correlate the information to VTAM and the communication controller by using the communication identifier (CID), and to VSCS and CCS using the IUCV path identifiers.

Note: The CONFTXT operand on the DTIGEN macroinstruction specifies whether the tracing of user buffer data should be suppressed. The default (CONFTXT=N) does not suppress tracing. See the description of the CONFTXT operand of the DTIGEN macroinstruction in the *VTAM Resource Definition Reference*.

Display Command

The VSCS DISPLAY command allows you to display multiple LUs in session with VSCS and determine the characteristics and status of the LUs. Use this information to resolve problems without additional documentation (for example, when an incorrect logon mode is used for an LU). Specifying this command without a network qualifier produces a display of all LUs with the specified LU name. To see LUs for only a particular network, use a network qualifier.

LU Dump Command

The VSCS DUMPLU command enables you to dump an LU's control blocks, trace table, and all associated work elements. If the number of storage pages to be dumped exceeds the limitations of GCS, then VTAM requests a full dump. If you provide multiple LU names, VTAM creates a separate dump for each LU and uses a dump name that reflects the LU that was dumped. Specifying this command without a network qualifier produces a dump for all LUs with the specified LU name. To see LUs for only a particular network, use a network qualifier.

Storage Command

The VSCS STORAGE command enables you to display dynamic storage utilization, user options, CP and VTAM session counts, and GETMAIN, FREEMAIN, and swap counts. If you do not use the STCHKTM and STRELTM storage options of DTIGEN, some of the messages will not be displayed.

Buffer Commands

The VSCS storage manager maintains queues of dynamic free buffers. If you specify queues on a FIFO basis, all the buffers are used in a cyclical manner before any are reused. Use this as a debugging aid to examine the contents of a buffer, not just a few bytes, as with the trace table. See the description of the BFRFIFO operand of the DTIGEN macroinstruction in the *VTAM Resource Definition Reference*. This value should be used only when required for serviceability because it requires additional path length.

The VSCS BFRLIFO command specifies the use of the queue of dynamic free buffers on a last-in-first-out basis. This is the default setting and the recommended value. BFRFIFO specifies the use of the queue of dynamic free buffers on a first-in-first-out basis for a storage full buffer trace.

Query Command

The VSCS QUERY command lets you check the status of the following diagnostic tool options:

- TRACE options
- Buffer queuing
- Location of the VSCS trace table, if one exists.

Halting VSCS

The VSCS component of VTAM for VM can be stopped independently of VTAM. The VSCS QUIT command stops VSCS execution. To simplify operations, VSCS accepts VSCS HALT as a synonym for the QUIT command. It might be necessary to cancel VSCS if VSCS cannot be terminated by QUIT or HALT processing. Canceling VSCS with the CANCEL command should be done only if HALT does not work. The CANCEL command causes the immediate abnormal termination of VSCS with the system completion code of X'13E' without any orderly cleanup of LUs.

Additional VSCS Considerations

This section contains considerations for the following:

- Printers
- Terminal logons
- Terminal logoffs
- VM logo usage
- Accounting information.

Printers

Printers are used by VSCS solely as local copy devices. An LU can define a CP PF key as SET PFnn COPY *netid.luname*. When a terminal user presses this PF key, the contents of the screen are sent to the printer specified by *netid.luname*. If the printer is defined in the VSCS printer table but is not currently in session with VSCS, a VTAM SIMLOGON is issued to acquire it. If *luname* is specified without a network ID, the host NETID is used.

Printers can be logged on to VSCS in the following ways:

- The VTAM operator can issue a VARY LOGON command.
- A VTAM definition statement can specify a printer that is controlled by VSCS.

If you want the printer to become active as soon as VTAM is started, include the major node name in which the printer resides in an ATCCONxx VTAMLST file. "Creating Start Option Lists" on page 130 describes ATCCONxx VTAMLST files.

- The VSCS operator can issue a VSCS PRINTER command.

In this case, no session is established between VSCS and the printer; the printer's LU name is added to the VSCS printer table. (The number of entries in the printer table is determined by the VSCS initialization parameter PRNTNUM.)

Then, when a display requests a copy, VSCS requests a session with the printer. If a session cannot be established within the number of seconds specified by the TIMECPY option, the display status area is changed to NOT AVAILABLE, and the copy request is terminated.

The VSCS PRINTER command allows printers to be shared among several VTAM application programs. Even different VM systems can share the same printer for PF key copy functions. If a printer has been requested by another application program, VTAM drives the VSCS RELREQ exit. When the printer is not in use and has not been used within the number of seconds specified by the TIMERELOption, its release depends on the following conditions:

- If PRTSHR=N (VSCS option), the printer is not released, and the operator must use the VTAM VARY INACT or the VSCS FORCE command to release the printer.
- If PRTSHR=Y (VSCS option), one of the following occurs:
 - If the printer was logged on using either the VTAM VARY command or a VTAM definition statement, the printer is released and the printer's LU name is removed from the printer table.
 - If the printer was added using the VSCS PRINTER command, the printer is released, but the printer's LU name is left in the printer table. This way the printer can be reacquired for the next PF-key copy request.

Another way to remove a printer is by using the VSCS FORCE command. If a printer is currently attached to VSCS, force processing releases the printer (using a VTAM CLSDST) and removes it from the printer table. If the printer is not currently attached or it is being logged on, force processing removes the printer from the printer table.

Note: If the logon mode of a printer does not indicate that the device is a printer or if the luname in the printer table does not represent a printer device, the device is logged on to VSCS as something other than a printer. Problems can result in attempting to clean up this situation because the original logon request for the printer never completed.

Terminal Logons

Because VSCS is a VTAM application program, terminal users must perform a VSCS logon before doing a CP logon to VM. However, the VSCS logon can be made transparent to terminal users. The examples of VSCS logons in this section assume that the network name of the application program (VSCS) is VM, which is the default. (The network name is the minor node name assigned to the application program. It is defined on the VTAM APPL definition statement.)

After you enter the following statements, the VM logo is displayed, and you can issue a CP logon.

```
LOGON APPLID(VM)
VM
```

The format of VSCS logons can be customized. In the preceding example, the VSCS logon is a USS command; it is defined in the default USS table. An alternate USS table can be coded to define additional forms of a VSCS logon. An entry in an interpret table can also be used to customize the format of a VSCS logon.

Terminal users can include CP logon information on the VSCS logon to eliminate the extra step of a CP logon. The CP information can be included as parameters on the VSCS logon. The following statements show CP information included on a VSCS logon:

```
LOGON APPLID(VM) DATA(userid)
LOGON APPLID(VM) DATA(userid options)
LOGON APPLID(VM) DATA('DIAL userid')
VM userid
VM userid options
VM DIAL userid
```

Any CP command valid before a CP logon can be used as data. However, the DIAL command is restricted to use only by 3270 display terminals. If DATA contains all blanks, it is ignored, and the VM prompt screen is displayed (if provided by CP).

Note: All data entered in DATA is re-displayed. To prevent re-display, you must code the appropriate VSCS data manipulation exit routines.

VSCS logon can be bypassed by automatically logging the terminal on to VSCS. A terminal's definition statement can specify automatic logon to VSCS using the LOGAPPL operand. This causes the terminal to receive the VM logo when it is powered on. This way the VSCS logon appears identical to a native CP logon. The VTAM operator can also use the VARY LOGON command to initiate a session with VSCS without intervention from the terminal user.

As mentioned previously, VSCS also allows the CP DIAL command for 3270 display devices. After a VSCS logon but before a CP logon, CP DIAL can be used to attach a terminal to another virtual machine. This method makes an SNA terminal appear as a non-SNA terminal to the virtual machine that was dialed. DIAL can be issued as user data in the VSCS LOGON command or as a CP command. Keyboard/prINTER devices such as the 2741, 3767, or 3101 and TWX terminals cannot use this command.

VM provides a display (3270 devices) logon facility, which allows terminal users to logon from the logo screen (without first clearing the logo). When the logo is displayed, the cursor is positioned at the first character of the command lines (*userid*).

Terminal Logoffs

When the terminal user logs off of CP, the user might or might not be logged off of VSCS. This is dependent on the way the terminal user logged on and the value of the VSCS option ONELOGO.

If a user has logged on to VM after logging on to VSCS (in separate steps), the CP LOGOFF, LOGON HERE, or DISCONNECT command results in the following:

- If ONELOGO=N, the user is not logged off of VSCS. Instead, the VM logo reappears, and the user might enter pre-logon commands, including logging on to VM again or typing VMEXIT to log off VSCS and return to VTAM control. (VMEXIT can be entered on any of the three command lines, and it overrides any other data entered.)
- If ONELOGO=Y, the user is logged off of VSCS and returned to VTAM control.

If a user has logged on to both VSCS and VM in one step, the CP LOGOFF or DISCONNECT command results in logging the terminal user off of VSCS and returning the user to VTAM control.

Users can also end sessions with VM by turning their terminals off and then on. VSCS disconnects the terminal when the power comes on again. If the power-off action does not generate an error condition or if power-on action does not generate an "unsolicited device end" condition, VSCS is not aware that a power-off action occurred. Therefore, it does not disconnect the LU from CP.

If a user of a keyboard/prINTER or TWX device has logged on to VSCS without supplying data (*userid*), the VM online message is displayed. At this point, the atten-

tion or break key must be pressed before the keyboard is unlocked. When logoff or disconnect is issued, the attention/break key must be pressed once to receive the VM online message and once more to log on again. The keyboard/printer can issue a VMEXIT after using the attention/break key with the VM online message displayed.

VM Logo Usage

Following are three possible VM display logos and what they consist of:

- Normal, also referred to as the standard logo:
 - Online message
 - Logo body
 - Command lines
- Single line, also known as the alternate logo:
 - Online message
 - Single line logo body
- Online message (consists of online message only).

For keyboard/printer and TWX LUs, only the online message is used.

The device characteristics (screen size) and the logo requirements (screen size) determine which one of the logos to use. When building a logo for a display, the screen width and the screen depth are compared against the normal logo. If the logo does not fit, the single line logo is evaluated, and if all else fails, only the online message is used. This should match the process used for native devices.

Figure 155 on page 646 shows the default logo. If you are using a customized logo that is much deeper than the default logo and if the command lines with the logo body do not fit on the screen, only the online message and a single line logo (alternate) are displayed, as shown in Figure 156 on page 646.

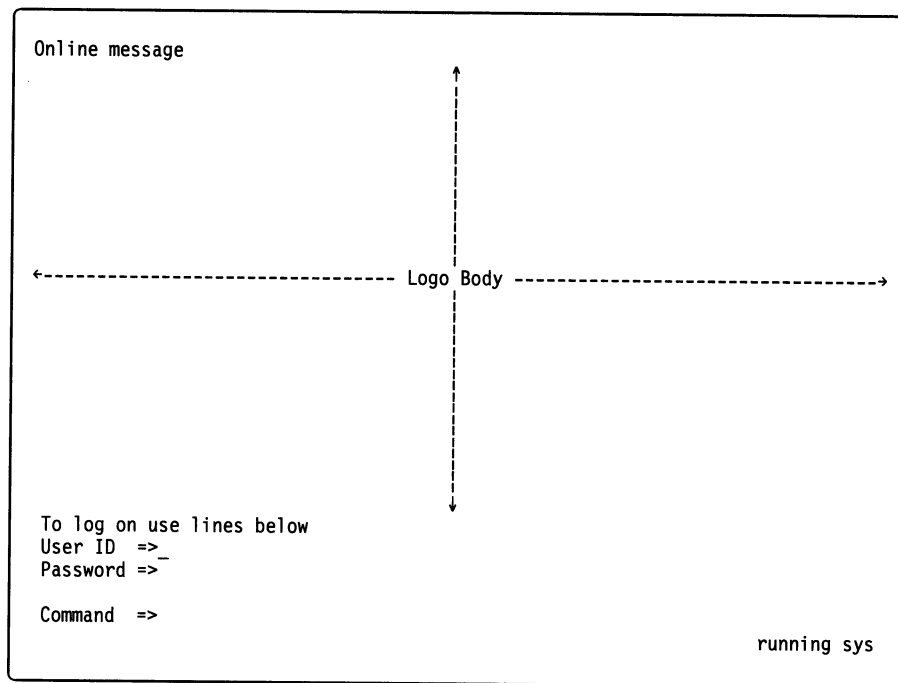


Figure 155. Default Normal Logo

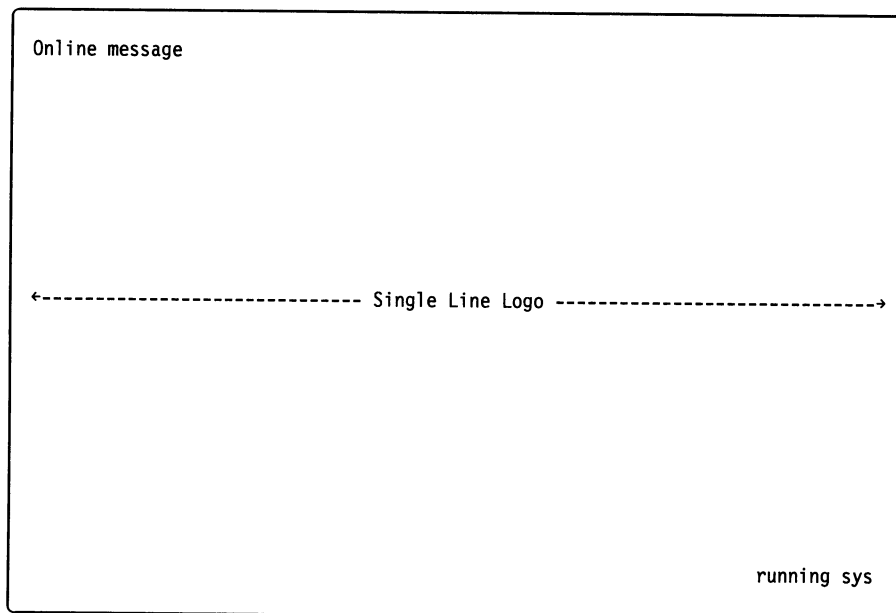


Figure 156. Single Line Logo

To ensure proper logo handling in VSCS, consider the following when you are customizing the logo:

- In most cases, your customized logo should appear the same on a VTAM display as it does on a CP controlled display. However, VSCS is more sensitive to the number of lines used in a logo, so remember the following:
 - Allow a single line for the online message.
 - Allow two lines for the input area.

- Compensate for the command lines (including the first line of the input area).
 - VSCS is also sensitive to the width of each line in the logo. Therefore, use a line width that is less than or equal to the maximum line width minus 2 (78 characters on an 80-character screen).
- Note:** Screen control commands are considered when determining the actual width that is required.
- It is strongly recommended that you do not modify the command lines because of VSCS's dependency on the format and content. However, if you must, remember the following points so that the logo can still support logons:
 - Keep the user ID, password, and command on separate lines.
 - Place the insert cursor (IC) at the first usable field of the user ID line (see Figure 155 on page 646).
 - Make sure each command line is bound by the appropriate attributes (that is, protected, masked, or unprotected).
 - If you imbed Start Field (SF) and Start Field Extended (SFE) commands in your logo, make sure they are supported by all VTAM devices that logon to VSCS.
 - All set buffer addresses and invalid (not supported) screen control commands are blanked out. This can distort the logo.

During VSCS initialization, operator awareness messages are issued to indicate the following:

- Normal logo body does not fit on a 24x80 screen:
 - LINES—too many lines exist.
 - WIDTH—one or more lines of the logo is too wide.
 - SIZE—both of the preceding are true.
 - Extended data stream commands are contained in the data.
- Logon from the logo screen is not supported if command lines are not available or are invalid. This does not prevent use of the supplied customized logo.

Accounting Information

VSCS presents accounting information to VM when a user logs off or disconnects. This data is recorded in a punch or spool file for user processing. Accounting data can assist in performance tuning of VSCS. Information in the VSCS accounting records can be used to calculate the average buffer sizes for VSCS send and receive operations. The results of these calculations can be used to decide optimal values for the VSCS start options KPXM TL, DPXM TL, and RCVBFRL.

CP adds a 16-byte prefix to the accounting record. This prefix contains the user ID or terminal identification, the accounting number, and a 2-byte suffix indicating the accounting record identification code (07 for VSCS). The information shown in Table 42 on page 648 is contained in columns 17–78 of the VSCS accounting record. Refer to the *VM/ESA Library Guide and Master Index* to locate the appropriate book for your system that contains more information on the accounting record.

Note: In some error situations, VSCS does not provide an accounting record, or CP discards it.

Table 42. VSCS Accounting Record Format

Record	Size (Bytes)	Description
Overflow counters	4	Number of times counters 1–4 reached their maximum values. ¹ on page 648 Byte Meaning 1 Overflow for counter #1 2 Overflow for counter #2 3 Overflow for counter #3 4 Overflow for counter #4
Counter #1	4	Number of input request units (RECEIVES)
Counter #2	4	Total number of input bytes
Counter #3	4	Number of output request units (SENDS)
Counter #4	4	Total number of output bytes
Counter #5	4	Number of console output lines
Counter #6	4	Number of copy requests
Logon time stamp	8	The time stamps are in the following format: ² on page 648 Time (4 bytes): <i>hhmmssst</i> Date (4 bytes): <i>00yydddC</i>
Logoff time stamp	8	The format is the same as that shown for the logon time stamp. ² on page 648
Logical unit name	8	The name of the logical unit logging off.
Network ID	8	The network qualifier for this LU.

Note:

1. The maximum value for each counter is 2147483647. To find the actual counter values, multiply the overflow count value times the maximum value, then add the amount in the counter.
2. The time stamp data is in packed decimal digits:
hh = hours (24-hour clock)
mm = minutes
ss = seconds
t = tenths of seconds
h = hundredths of seconds
yy = the last two digits of the year
ddd = the day of the year
C = a 4-bit sign character that allows the data to be unpacked and printed.

Appendix C. Tuning VM for VTAM

This appendix describes how to tune VM systems for VTAM.

Throughout this appendix, you might need to refer to the following books:

- *VM/ESA Operator's Guide*
- *VM/ESA CP Planning and Administration*

The following sections describe options that improve the performance of VTAM in VM environments. These options make VTAM the highest dispatchable task under CP's control. The heading for each section indicates what systems the section applies to.

SET RESERVE

RESERVE is an operand of the CP SET command and requires privilege class A.

SET RESERVE specifies the number of page frames that are reserved for the specified virtual machine's exclusive use (in this case, VTAM's). The number of frames held is either the value specified or the actual amount of dynamic storage, whichever is smaller.

To achieve the maximum VTAM throughput, the paging rate on behalf of the VTAM virtual machine must approach zero, meaning that all of VTAM's instructions are in real storage at all times. However, reserving too many pages (and setting the minimum dynamic storage value too high) can adversely impact other system users. Following is a description of two software tools that can monitor the real storage utilization:

- VM realtime monitor takes a snapshot of the system, displaying the system's most active users and the amount of paging taking place on behalf of the users displayed.
- VM MAP reports the activity on behalf of individual virtual machines. If the paging rate for the VTAM service machine (VSM) is consistently greater than one per second, the minimum amount of dynamic storage and amount of storage reserved should be increased.

DIAG98

The DIAG98 option allows VTAM to translate storage addresses in its CCW strings from virtual addresses to real addresses. Because VTAM uses specialized code for the translation, this normally uses less CPU time than CP's generalized translation.

Page Locking

VTAM needs to lock pages for several uses, I/O buffers, NCB's for channel attached devices and other control blocks for channel end processing.

VTAM's use of DIAG98 is limited to the locking of its I/O buffers and control blocks containing CCWs. The amount of storage locked is determined by the allocation of IOBUF in the ATCSTRxx VTAMLST file. This storage is locked during initialization

and is not unlocked until VTAM is halted. The control blocks containing CCWs are locked by VTAM from various pools. The locking process is controlled internally by VTAM and typically takes place when a physical unit is activated.

When DIAG98 is enabled, VTAM uses it for all I/O to NCP, local SNA controllers, channel-to-channel attachments, and local non-SNA, non-dialed 3270 devices. VTAM does not use DIAG98 facilities for devices that are CP dialed to the VTAM machine (as some local non-SNA 3270s are). These devices are usually defined to VTAM using the SPECIAL directory statement.

This option can be selected at system generation time using the DIAG98 operand of the OPTION control statement in the virtual machine's entry in the system directory. The recommendation for VTAM is as follows:

```
USER VTAM passwd ...  
OPTION.....DIAG98
```

XEDIT SET REMOTE

XEDIT SET REMOTE enables all terminals that access the system through VSCS, IUCV, or CCS to appear as channel-attached terminals. If an application program were to interrogate VM control blocks (RDEVBLK) to obtain a description of the device, it would find each VTAM terminal defined as a local device, even though the device might actually be remote.

XEDIT is an example of an application program that interrogates this block to determine whether the device is local or remote. If the device is local, the full data stream is transmitted to the terminal. If, however, the device is remote, XEDIT compresses the data stream. This compressed data stream decreases the amount of data that is actually transmitted on the line, thereby improving the response time and line utilization.

A user can override the automatic setting by the XEDIT subcommand SET REMOTE ON when in XEDIT, or from the XEDIT profile.

The recommendation for remotely attached VTAM terminals using XEDIT is as follows:

```
SET REMOTE ON
```

See the *VM System Product Editor Command and Macro Reference* for more information.

Note: SET REMOTE ON is a full-screen CMS command in CMS SP5.

SET QUICKDSP

SET QUICKDSP performs a function similar to SET PRIORITY. The use of the QUICKDSP operand causes the VTAM virtual machine to be added to the dispatching list immediately when it has work to perform, without waiting in the eligible list. It is recommended that you use SET QUICKDSP.

Refer to *VM/ESA CP Planning and Administration* for more information.

SET SHARE RELATIVE 10000

SET SHARE RELATIVE 10000 enables you to control the percentage of system resources that a virtual machine receives. These system resources include the processor (CPU cycles), real storage, and paging I/O capability. The RELATIVE operand specifies the relative share of the system assigned to a virtual machine. The maximum value that can be specified is 10000. Therefore, the VTAM virtual machine is allocated the maximum portion of total system resources available. It is recommended that you specify SET SHARE RELATIVE 10000.

Refer to *VM/ESA CP Planning and Administration* for more information.

Appendix D. Logon Manager (MVS)

The transaction processing facility (TPF) is an application program dedicated to high-speed transaction processing. It is designed to handle a large volume of transactions submitted from many remote logical units. The logon manager is a VTAM application program that provides logon services for TPF.

The logon manager provides logon access to a TPF application program from independent and dependent logical units. If you define your TPF application program as a host processor (type 5), only dependent LUs can access the TPF application program. However, if you define your TPF application program as a type 2.1 physical unit, both dependent and independent LUs can access the TPF application program.

Note: To implement the logon manager, you need to have NCP V4R3 or a subsequent release.

This appendix contains the following sections:

- How the Logon Manager Operates
- Installing the Logon Manager
- Creating JCL for the Logon Manager
- Defining the Logon Manager.

How the Logon Manager Operates

How the logon manager receives control depends on whether a dependent or independent LU is accessing the TPF application program. If the logon request is from a dependent SLU, the logon manager receives control when VTAM drives its logon exit. If the logon request is from an independent PLU, the logon manager receives control when VTAM drives the USERVAR exit, ISTEEXCUV. ISTEEXCUV is driven because logon manager creates a USERVAR (with UVEXIT=YES) to represent the generic name of each TPF application program.

After logon manager receives control, the control point LUs that support the generic application are identified. Each pair of supporting control point LUs is then passed to the CLU search exit, ELMCLUEx, to determine the LU best able to provide the service. The selection criteria of the ELMCLUEx exit can be determined by the user or the default can be used. The default criteria are:

- LU is below the session limit
- LU has the lowest hop count to the device
- LU has the fewest number of active sessions.

For dependent LU access, the TPF application program can be defined as either a type 5 or type 2.1 PU.

For independent LU access, the TPF application program must be defined as a type 2.1 PU.

Installing the Logon Manager

The VTAM installation process places the logon manager load modules in SYS1.LINKLIB. You can put the logon manager in another load library by modifying the installation process. If you do this, you must either add this other load library to your system's link list or point the logon manager JCL to this library by using a STEPLIB DD statement. Also, logon manager must be APF authorized.

To run the logon manager, you must define the logon manager program ELMNGR as nonswappable with a protection key value of 6. To do this, update the MVS program properties table. See the SCHEDxx macroinstruction in *MVS/ESA System Programming Library: Initialization and Tuning Reference*.

Starting the Logon Manager

To start the logon manager, issue the START command START *procname*, where *procname* is the name of the logon manager JCL procedure. The logon manager can also run as a job. If you want to run it as a job, modify the logon manager JCL and the start procedure to conform to your installation's standards for such jobs.

You must place JCL for the logon manager in SYS1.PROCLIB or an equivalent library with a member name *procname* of your choice. Following is a sample of the required JCL:

```
//name      PROC MEMBER=partitioned data set member name,
              MAXSUBA=value
//GO        EXEC PGM=ELMNGR,PARM='MEMBER=&MEMBER,MAXSUBA=&MAXSUBA'
//ELMDEFDS DD DSN=partitioned dataset name,DISP=SHR
```

Note: The quotes are required around the PARM field of the EXEC statement.

name

is optional. If specified, it is usually the same as the member name *procname* of this JCL procedure.

ELMNGR

is the name of the logon manager load module.

ELMDEFDS

is the required DD name of the DD statement for the partitioned data set that contains the logon manager configuration definition.

MEMBER

is the 1–8 character name of the member of the ELMDEFDS partitioned data set that contains the logon manager configuration definition.

MAXSUBA

specifies the maximum number of network subareas that can be defined to the logon manager. Valid values are 1–511. The default is 1. The number specified must be at least as great as the number of network subareas defined in the logon manager configuration definition.

Defining the Logon Manager

This section describes how to define the logon manager and includes a description of a typical logon manager configuration.

Note: You can also define TPF as a type 5 data host, rather than a type 2.1 physical unit. As a data host, TPF supports only fixed pacing window sizes, virtual route 0 (VR=0), and transmission priority 2 (TP=2). The TPF data host must reside in its own SNI network with VTAM in another host acting as the gateway VTAM. In this type of configuration, you need to code the TPF application programs as dependent logical units. These are the only resources the TPF data host can own. When TPF is a type 5 data host, logon manager should not be used.

Sample Logon Manager Configuration

Figure 157 on page 656 shows a logon manager configuration consisting of the following:

- TPF with control point LUs
- TPF defined as a type 2.1 PU
- VTAM with the logon manager
- Dependent SLUs.

NCP1 through NCP3 are NCP V4R3 subarea nodes, which are channel-attached to TPF. NCP3 is also channel-attached to VTAM. NCP1 through NCP6 are link-attached to each other. NCP4 through NCP6 do not have to be NCP V4R3 or subsequent releases.

CLU1 through CLU3 are control logical units (CLUs). They manage the flow of transactions from their associated subarea nodes.

TPF is defined to VTAM as a type 2.1 peripheral node. Because the NCPs are channel-attached, TPF requires LNCTL=CA on the NCP GROUP definition statement. Interactions between logon manager and TPF occur in the form of control point LU to control point LU sessions.

APPLA and APPLB are TPF application programs. They are defined to VTAM as local applications.

Resources defined as independent LUs can initiate sessions with the TPF applications.

To take advantage of multiple-channel attachments into the subarea network, each application program (for example, APPLA) needs at least one LU per channel-attachment. In this configuration, each application program requires three or more LUs (for example, APPLA1, APPLA2, APPLA3) to use all channels into the network. Because APPLB has only two LU definitions, it can use only two channels into the network.

SLU1 and SLU2 are defined to VTAM as dependent SLUs. They are link-connected to their associated subarea nodes and represent the remote terminal users of the TPF application programs.

You can define class-of-service (COS) tables for a subarea configuration. For information about COS tables, see "Class of Service (COS)" in the *VTAM Resource Definition Reference*.

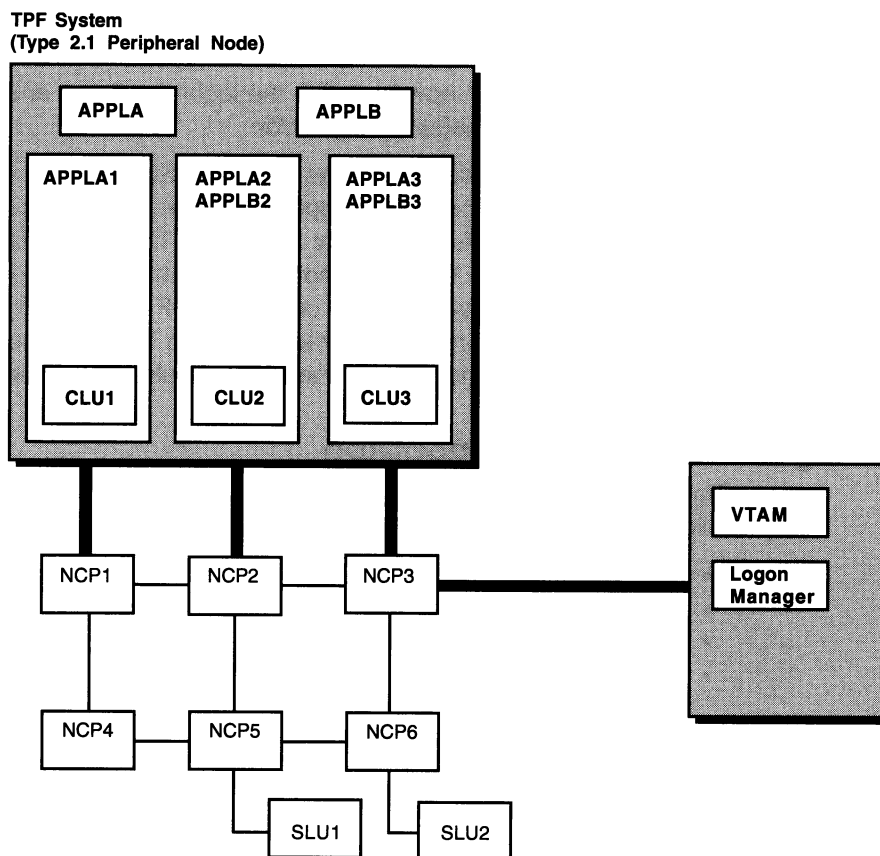


Figure 157. Sample Logon Manager Configuration

Defining the Logon Manager and TPF Applications to VTAM

You need to define the logon manager and each TPF application program to VTAM using APPL definition statements. The following operands and operand values are required:

```
name      APPL  ACBNAME=acbname,
              AUTH=(ACQ,NOCNM,NOPASS,NOTSO,SPO),
              PARSESS=YES,
              SRBEXIT=NO,
              SONSCIP=YES,
              ENCR=NONE,
              VTAMFRR=NO,
              HAVAIL=YES
```

If *acbname* differs from the name *procname* of the logon manager JCL, you need to code the LMPROC definition statement in the logon manager configuration definition.

The following APPL definition statement operands are optional:

```
PRTCT
USSTAB
VPACING.
```

To use password protection for the logon manager, include the PRTCT operand on the APPL definition statement for the logon manager, and specify the same password as that on the LMPROC definition statement in the logon manager configura-

tion definition. To use password protection for a TPF application program, include the PRTCT operand on the APPL definition statement for the TPF application program, and specify the same password as that on the LMAPPL definition statement in the logon manager configuration definition.

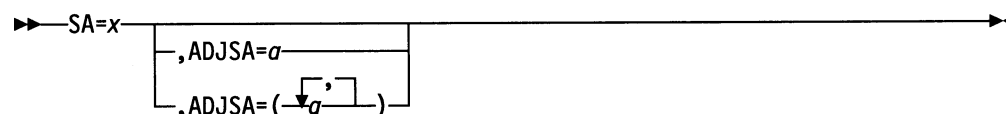
Defining the Logon Manager Configuration

The logon manager configuration is defined by a sequence of source statements stored in a member of a partitioned data set with the member name that is specified in the logon manager JCL. You can write different types of source statements in any order in the configuration definition. Additional logon manager configuration definitions can be stored in members with other member names.

A logon manager configuration definition always contains one SA definition statement identifying each network subarea. Other logon manager configuration definition statements are optional.

Subarea Configuration

The system programmer responsible for identifying the network subareas and their adjacent subareas must provide a configuration of the network subareas. Each network subarea must be defined in the logon manager configuration definition using one of the following formats:



x is the address for a subarea (SA).

a is the address for an adjacent subarea (ADJSA). The subarea must be adjacent to subarea **x**.

You can specify multiple adjacent subarea addresses by enclosing them within parentheses and separating them with a comma.

All network subareas that you are defining to the logon manager must have their addresses appear in SA definition statements as either **x** or **a**.

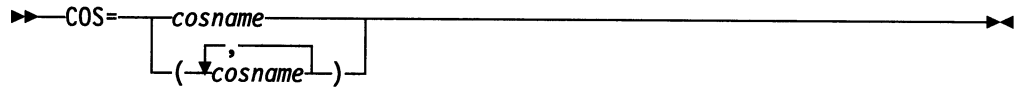
Note: The value of MAXSUBA specified on the EXEC statement in the logon manager JCL must be at least as great as the number of different network subareas addressed as **x** or **a** in the SA definition statements in the logon manager configuration definition.

The SAs supplied by the system programmer are used by the logon manager to build a HOP table. The HOP table contains the minimum distance between subareas and is used for multitask support.

Class-of-Service Configuration

With logon manager, you can use class of service (COS) to distinguish between the needs of various sessions. Logon manager differentiates between session types based on the COS name. By designating different COS names for batch and interactive traffic (or any other preferred distinction), you can keep the various types of traffic from interfering with each other. Using this function, you define multiple views of the network that list preferred paths for different types of session traffic.

To specify the COS name, use the COS definition statement in the following format:



For example in Figure 157 on page 656, NCP6 is only 1 hop away from NCP3, but is 2 hops away from NCP2. You can then specify that NCP3 is the preferred boundary function for any interactive sessions to the host and that NCP2 is the preferred route for batch traffic. To define the different routes to logon manager, you can define a number of classes of service, each with its view of the network. In the previous example you would code one class of service for interactive traffic, and one for batch.

To define these different views, use the COS definition statement. For example, to code the interactive view described previously, you would code the following:

```

COS=INTERACT
SA=6, ADJSA=(3)
  
```

Using this technique, NCP3 would be used for interactive traffic to the host.

The result is that each session is assigned to a TPF server based on predefined routing requirements, and enables you to assign suitable routes.

The SA definition statements that follow a COS definition statement apply only to the table representing that COS. If you code an SA definition statement without a COS definition statement preceding it, that SA definition statement is used as part of the default table.

If a COS name is not found for a COS that is requested, logon manager uses the default tables for the session:

- If no COS name is specified for the session, the default table is used.
- If a COS name is for the session, but cannot be found, the default table is used.
- If the default table is to be used, but is not defined to logon manager, an error is reported.

In the following example, coded for the configuration in Figure 157 on page 656, the first entry is used as part of the default table, and the other entries are used for interactive and batch directed traffic.

```

SA=1, ADJSA=(2)
COS=INTERACT
SA=6, ADJSA=(3)
SA=5, ADJSA=(2)
COS=BATCH
SA=6, ADJSA=(2)
SA=5, ADJSA=(3)
  
```

Application Passwords and Minimum Channel Links

To define password protection for a TPF application program, include the following definition statement in the logon manager configuration definition. You can also use this definition statement to define the number of active channel links required before the application program is supported by the logon manager.

```

▶▶ LMAPPL=application program name
    └──,PASSWD=password──┘
    └──,MINLINK=value──┘
▶▶
  
```

application program name

is the name of the TPF application program. It must be the name of a real application program.

password

is the 1–8 character password for the TPF application program. Passwords are optional.

If password protection is required for a TPF application program, the password must be specified on the PASSWD operand on the LMAPPL definition statement and on the PRTCT operand on the VTAM APPL definition statement. There is no default password.

value

defines the minimum number of channel links that must be active before the application program can engage in a session. Valid values are 1–255. The default is 1. This definition is optional.

If all application programs on the TPF have the same password and the same required number of active channel links, you can use the following definition statement:

```

▶▶ LMAPPL=ALL
    └──,PASSWD=password──┘
    └──,MINLINK=value──┘
▶▶
  
```

Logon Manager Procedure

If the logon manager requires password protection or if the logon manager ACBNAME on the APPL definition statement differs from the MVS *procname* or *jobname*, include the following definition statement in the logon manager configuration definition:

```

▶▶ LMPROC=acbname
    └──,PASSWD=password──┘
▶▶
  
```

acbname

is the 1–8 character name of the logon manager's ACB. If you do not supply the LMPROC definition statement, the MVS *procname* or *jobname* is used.

password

is the 1–8 character password for the logon manager. The logon manager uses this password when opening the ACB for the *procname* or *jobname*. The logon manager password is optional.

If password protection is required for the logon manager, the password must be specified on the PASSWD operand on the LMPROC definition statement and

on the PRTCT operand on the VTAM APPL definition statement. There is no default password.

Number of TPF Applications

To specify the number of TPF application programs that can be defined, include the following definition statement in the logon manager configuration definition:

►►MAXAPLC=*value*◄◄

value

is the number of TPF application programs that can be defined. Valid values are 1–4095. The default is 4. This definition is optional.

Maximum Number of Buffers

To specify the maximum number of buffers to allocate for each channel-attached control point LU, include the following definition statement in the logon manager configuration definition:

►►MAXBUF=*value*◄◄

value

is the maximum number of buffers to allocate for each control point LU. Valid values are 2–65535. The default is 30. This definition is optional.

Logon manager takes the value specified in the MAXBUF definition statement and multiplies it by the value specified in the CHANATT definition statement to determine the total number of buffers available to logon manager. There is no restriction to the number of buffers that can be used for a given control point LU as long as there are free buffers available.

Maximum Number of Subareas

To specify the maximum number of channel-attached subareas, include the following definition statement in the logon manager configuration definition:

►►CHANATT=*value*◄◄

value

is the maximum number of channel-attached subareas. This number is used to determine the maximum number of channel-attached control point LUs. Valid values are 1–255. The default is 4. This definition is optional.

Reserved Keywords

The following keywords are reserved for use with the TPF logon manager:

- ADJSA
- ALL
- APPL
- CHANATT
- CLU
- HELP
- ID
- INFO
- LMAPPL
- LMPROC
- MAXAPLC

- MAXSUBA
- MEMBER
- MINLINK
- PASSWD
- PEND
- SA
- STOP.

Monitoring Logon Manager Resources

The VTAM logon manager application offers commands to monitor the status of the logon manager and the resources under control of the logon manager.

By using the logon manager MODIFY command, you can display information about logon manager resources (TPF applications, CLUs, and LUs), including session status. For more information on the logon manager operator commands, see Chapter 3, “Logon Manager Operator Commands (MVS)” in *VTAM Operation*.

Halting the Logon Manager

The logon manager functions as a stand-alone VTAM application and, thus, can be stopped independently of VTAM. The logon manager MODIFY STOP command stops logon manager processing. For more information on the logon manager operator commands, see Chapter 3, “Logon Manager Operator Commands (MVS)” in *VTAM Operation*. When the command is issued, the logon manager shuts down its subtasks, closes its ACBs, and returns control to MVS. Current sessions are allowed to end, but new sessions are not permitted.

The VTAM HALT, HALT QUICK, and CANCEL commands terminate the logon manager, as they do any VTAM application.

Appendix E. Cryptographic Keys (MVS)

If you use the VTAM data encryption facility, you need to file cryptographic keys on the cryptographic key data set at the appropriate host processors. For information on which hosts require cryptographic keys, see “Encryption Facility” on page 168.

This appendix describes how to file these keys for different cryptographic facilities for both single-domain and multiple-domain sessions. The available cryptographic services are:

- IBM Programmed Cryptographic Facility program product (PCF)
- IBM Cryptographic Unit Support program product (CUSP)
- IBM Integrated Cryptographic Service Facility/MVS (ICSF/MVS).

Note: Throughout this section, the acronym PCF/CUSP represents PCF, CUSP, or any compatible cryptographic product. However, specific commands and control statements for key input may differ by product.

For more information on establishing cryptographic sessions, refer to “How VTAM Determines the Level of Cryptography for a Cryptographic Session” in *VTAM Programming*.

Filing SLU Keys for Single-Domain Cryptographic Sessions

The process for filing SLU keys for single-domain cryptographic sessions varies, depending on which cryptographic service you use.

Single-Domain Cryptographic Sessions That Use PCF/CUSP

To allow cryptographic sessions that use PCF/CUSP to be established within a single domain, install either PCF or CUSP. To install PCF, refer to the *OS/VS1 OS/VS2 MVS Programmed Cryptographic Facility Installation and Reference*. To install CUSP, refer to the *Cryptographic Unit Support Installation Reference Manual*.

Use PCF/CUSP to file secondary logical unit (SLU) keys on the cryptographic key data set (CKDS) as follows:

- For each device-type LU that is to be used as the secondary end of a cryptographic session, code the following, where *name* is the name of the LU:

LOCAL *name*

This LOCAL statement generates an SLU key for the LU and adds it to the CKDS enciphered under the first variant of the host master key. It also returns a clear SLU key. Enter the clear SLU key into the device.

- For each VTAM application program that is to be the secondary end of a cryptographic session, code the following, where *name* is the name of the application program:

REMOTE *name*

This REMOTE statement generates an SLU key for the application program and adds it to the CKDS enciphered under the second variant of the host master key.

Figure 158 on page 664 illustrates the possible coding for cryptographic sessions that use PCF/CUSP. APPL1A can initiate a cryptographic session with LU1A. T1 can initiate a cryptographic session with APPL1A. However, APPL1A cannot initiate a session with T1.

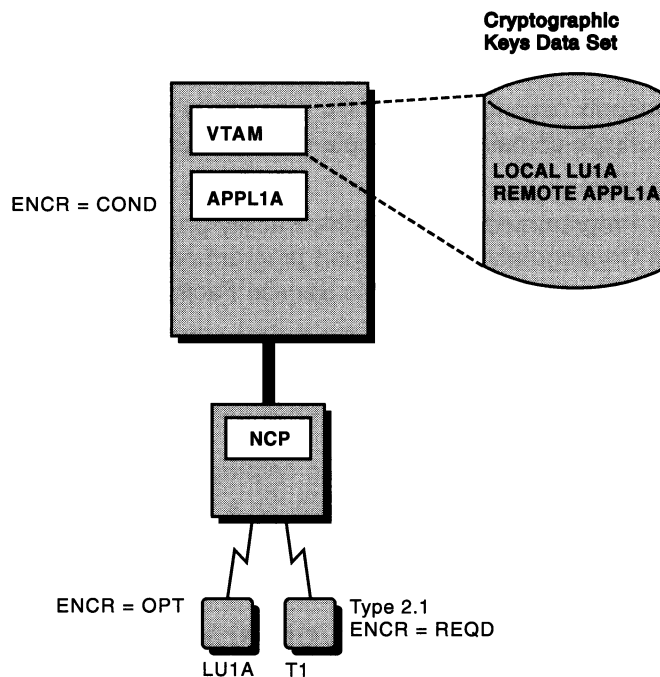


Figure 158. Cryptography in Single-Domain Environment That Uses PCF/CUSP

Single-Domain Cryptographic Sessions That Use ICSF/MVS

To allow cryptographic sessions that use ICSF/MVS to be established within a single domain, install ICSF/MVS using the *Integrated Cryptographic Service Facility/MVS Program Directory*.

Use ICSF/MVS to file secondary logical unit (SLU) keys on the cryptographic key data set (CKDS) as follows:

- For each device-type LU that is to be used as the secondary end of a cryptographic session, code the following, where *name* is the name of the LU:
ADD LABEL(*name*) TYPE(EXPORTER) CLEAR SINGLE

Note: SINGLE is needed only if the cryptographic support of the device-type LU does not support double length keys.

This key generation utility program (KGUP) statement generates an exportable SLU key for the LU and adds it to the CKDS that is enciphered under the host master key. It also returns a clear SLU key. Enter the clear SLU key into the device.

- For each VTAM application program that is to be used as the secondary end of a cryptographic session, code the following, where *name* is the name of the application program:
ADD LABEL(*name*) TYPE(IMPORTER) CLEAR SINGLE

Note: SINGLE is needed only if the cryptographic support of the device-type LU does not support double length keys.

This KGUP statement generates an importable SLU key for the application program and adds it to the CKDS that is enciphered under the host master key.

Figure 159 illustrates the possible coding for cryptographic sessions that use ICSF/MVS. APPL1A can initiate a cryptographic session with LU1A. T1 can initiate a cryptographic session with APPL1A. However, APPL1A cannot initiate a session with T1.

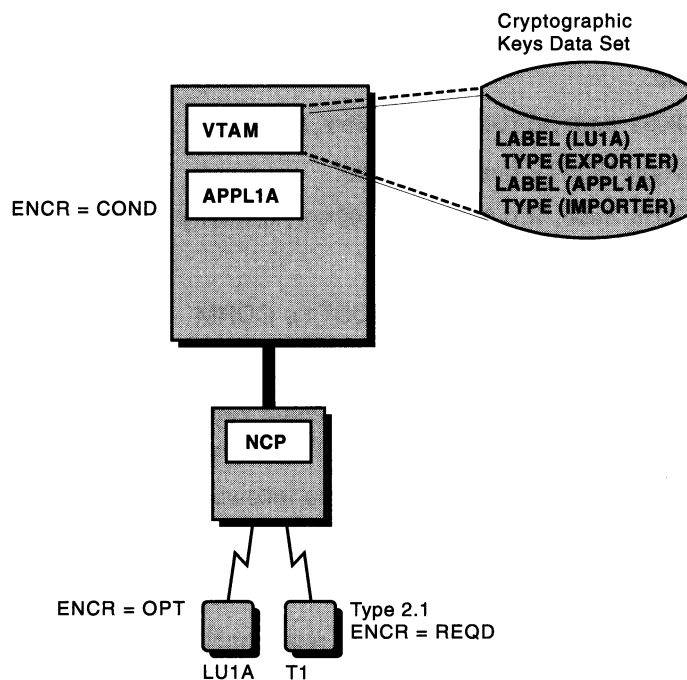


Figure 159. Cryptography in Single-Domain Environment That Uses ICSF/MVS

Filing CDRM Keys for Cross-Domain Cryptographic Sessions

The process for filing SLU keys for cross-domain cryptographic sessions varies, depending on which cryptographic service you use and whether both hosts use the same cryptographic service.

Cross-Domain Cryptographic Sessions Where Both Hosts Use PCF/CUSP

To allow cross-domain cryptographic sessions to be established, file SLU keys for each domain as described in "Single-Domain Cryptographic Sessions That Use PCF/CUSP" on page 663. Start PCF/CUSP before you activate the external CDRM for which cross statements have been filed in the cryptographic keys data set. Use PCF/CUSP to file cross-domain keys on the cryptographic key data set (CKDS) at each host processor as follows:

- For each pair of host processors (HOST1 and HOST2) that are to have cross-domain cryptographic sessions between their domains, code the following at HOST1, where *name* is the name of HOST2's CDRM:

CROSS *name*

This CROSS statement generates two cross-domain keys, one defined as local and the other defined as remote.⁵ It stores the local cross-domain key in the cryptographic key data set (CKDS) enciphered under the first variant of HOST1's host master key and stores the remote cross-domain key in the CKDS enciphered under the second variant of HOST1's host master key. Both of these keys are associated with the name of HOST2's CDRM. This CROSS statement also returns clear copies of the two keys.

- The cross-domain keys generated at HOST1 must be used at HOST2 and supplied as input to PCF/CUSP in a CROSS statement. When HOST2 has PCF use:

```
CROSS name,KEYLOC=x,KEYREM=y,ADD
```

When HOST2 has CUSP use:

```
CROSS name,KEYLOC=x,IKEYLOC=xx,KEYREM=y,IKEYREM=yy,ADD
```

Where:

name is the name of HOST1's CDRM

x is the clear remote cross-domain key from HOST1

xx is the second half of double key value from HOST1's remote key

y is the clear local cross-domain key from HOST1

yy is the second half of double key value from HOST1's local key

This CROSS statement does the following:

- Adds the two cross-domain keys to HOST2's CKDS
- Associates the two keys with the name of HOST1's CDRM
- Reverses the local/remote relationship of the two keys
- Enciphers key *x* and its intermediate key *xx* under the first variant of HOST2's host master key, and key *y* and its intermediate key *yy* under the second variant of HOST2's host master key

Figure 160 on page 667 illustrates the possible coding for cross-domain cryptographic sessions where both hosts use PCF/CUSP. APPL2A can initiate a cryptographic session with LU1A. APPL1A can initiate a cryptographic session with LU2A. APPL2A can initiate a cryptographic session with APPL1A. However, APPL1A cannot initiate a session with APPL2A because there is no REMOTE statement coded in VTAM2 and APPL2A requires cryptography.

⁵ The terms *local* and *remote* in reference to the keys used by the CROSS statement do not have the same meaning here as they do in other contexts in this book. For more information on PCF, refer to the *OS/VS1 OS/VS2 MVS Programmed Cryptographic Facility Installation and Reference*. For more information on CUSP, refer to the *Cryptographic Unit Support Installation Reference Manual*.

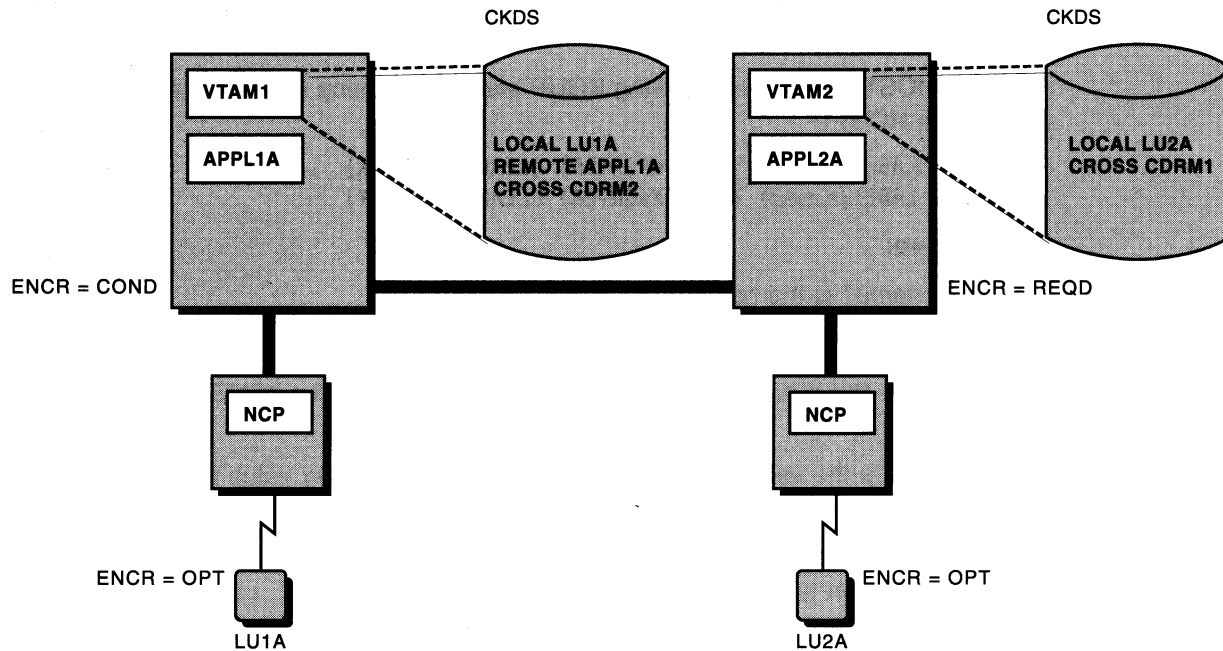


Figure 160. Cryptography in Multiple-Domain Environment (Both Hosts Use PCF/CUSP)

For the configuration in Figure 160 to have any encrypted sessions, start cryptography in both hosts before activating a session.

Cross-Domain Cryptographic Sessions Where Both Hosts Use ICSF/MVS

To allow cross-domain cryptographic sessions to be established, file SLU keys for each domain as described in "Single-Domain Cryptographic Sessions That Use ICSF/MVS" on page 664; then use ICSF/MVS to file cross-domain keys on the cryptographic key data set (CKDS) at each host processor as described in the following paragraphs.

A complementary pair of exporter and importer keys must be generated for the two host processors (HOST1 and HOST2). To allow for cross-domain cryptographic sessions between hosts using the ICSF/MVS cryptographic service, perform the following steps:

- At HOST1, code the following key generation utility program (KGUP) statements:

```
ADD LABEL(name1) TYPE(EXPORTER) CLEAR
ADD LABEL(name2) TYPE(IMPORTER) CLEAR
```

Where:

name1 is the name of HOST2's CDRM
name2 is the name of HOST1's CDRM

These KGUP statements generate an exporter key for HOST1 to use to encrypt session keys sent from HOST1 to HOST2 and an importer key for HOST1 to decrypt session keys sent from HOST2 to HOST1. The exporter and importer keys are placed in HOST1's CKDS. The statements also cause the clear key values to be placed in the KGUP key output data set. Assume clear key value X,XX for the exporter key and clear key value Y,YY for the importer key.

These values are to be used on HOST2 to build complement key control statements for these keys.

- At HOST2, code the following KGUP statements using the clear key values from HOST1:

ADD LABEL(*name1*) TYPE(IMPORTER) CLEAR KEY(X,XX)

ADD LABEL(*name2*) TYPE(EXPORTER) CLEAR KEY(Y,YY)

Where:

name1 is the name of HOST2's CDRM

name2 is the name of HOST1's CDRM

X is the first half of the double-length clear key value of HOST1's exporter key

XX is the second half of the double-length clear key value from HOST1's exporter key

Y is the first half of the double-length clear key value from HOST1's importer key

YY is the second half of the double-length clear key value from HOST1's importer key

These control statements place these keys in HOST2's CKDS.

Figure 161 illustrates the possible coding for cross-domain cryptographic sessions where both hosts use ICSF/MVS. APPL2A can initiate a cryptographic session with LU1A. APPL1A can initiate a cryptographic session with LU2A. APPL2A can initiate a cryptographic session with APPL1A. However, APPL1A cannot initiate a session with APPL2A because there is no TYPE(IMPORTER) statement coded in VTAM2 and APPL2A requires cryptography.

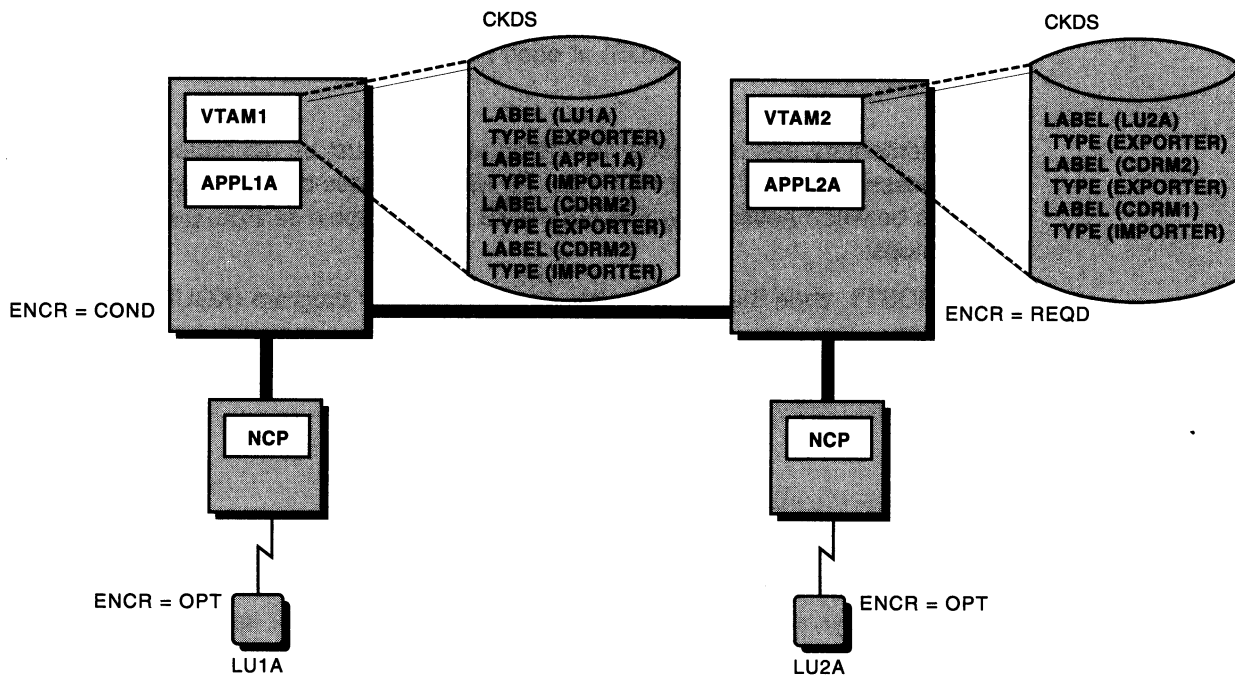


Figure 161. Cryptography in Multiple-Domain Environment (Both Hosts Use ICSF/MVS)

For the configuration in Figure 161 to have any encrypted sessions, start cryptography in both hosts before activating a session.

For more information on ICSF/MVS, refer to the *Integrated Cryptographic Service Facility Administrator's Guide*.

Cross-Domain Cryptographic Sessions Where the Hosts Use Different Cryptographic Services

In this description of possible coding, assume that ICSF/MVS is installed in HOST1 and PCF/CUSP is installed in HOST2. A complementary pair of key-encrypting keys must be generated for HOST1 and HOST2. To allow for cross-domain cryptographic sessions between the hosts, perform one of the following two sets of steps, depending upon where key processing is to be performed first:

- When key processing is to be performed from HOST2 first, perform these steps:

- At HOST2 where PCF/CUSP is installed, code the following key generation utility program (KGUP) statement:

```
CROSS name
```

Where *name* is the the name of HOST1's CDRM.

This PCF/CUSP KGUP statement generates a LOCAL and REMOTE key pair and places them in HOST2's cryptographic key data set (CKDS). A copy of the key pair generated will also appear in the KGUP SYSPRINT output listing. This gives HOST2 the capability to send encrypted session keys to HOST1.

- At HOST1, use the PCF/CUSP SYSPRINT output from the cross key pair to create the following ICSF control statement for the appropriate environment. When HOST2 has PCF:

```
ADD LABEL(name) TYPE(IMPORTER) CLEAR NOCV KEY(key-value)
```

When HOST2 has CUSP:

```
ADD LABEL(name) TYPE(IMPORTER) CLEAR NOCV KEY(key-value,ikey)
```

Where:

name is the name of HOST2's CDRM

key-value is the key value from HOST2 of the local key

ikey is the intermediate key value from HOST2 of the intermediate local key

This ICSF KGUP statement places the importer key, which is the same clear value as HOST2's LOCAL key in HOST1's CKDS. This allows HOST1 to decrypt session keys sent from HOST2.

- At HOST1, use the PCF/CUSP SYSPRINT output from the cross key pair to create the following ICSF control statement for the appropriate environment. When HOST2 has PCF:

```
ADD LABEL(name) TYPE(EXPORTER) CLEAR NOCV KEY(key-value)
```

When HOST2 has CUSP:

```
ADD LABEL(name) TYPE(EXPORTER) CLEAR NOCV KEY(key-value,ikey)
```

Where:

name is the name of HOST2's CDRM

key-value is the key value from HOST2 of the remote key

key is the intermediate key value from HOST2 of the intermediate remote key

This ICSF KGUP statement places the exporter key which has the same clear value as HOST2's REMOTE key in HOST1's CKDS. This allows HOST1 to send encrypted session keys to HOST2.

- When key processing is to be performed from HOST1 first, perform these steps:
 - At HOST1 where ICSF/MVS is installed, code the appropriate key generation utility program (KGUP) statements listed below for the environment in HOST2, where *name* is the the name of HOST2's CDRM. When HOST2 has PCF:

```
ADD LABEL(name) TYPE(EXPORTER) CLEAR NOCV SINGLE  
ADD LABEL(name) TYPE(IMPORTER) CLEAR NOCV SINGLE
```

When HOST2 has CUSP:

```
ADD LABEL(name) TYPE(EXPORTER) CLEAR NOCV  
ADD LABEL(name) TYPE(IMPORTER) CLEAR NOCV
```

These statements cause two keys to be generated in HOST1's CKDS. One key, the exporter, is to be used when sending session keys to HOST2. The other key, the importer, is to be used to decrypt session keys sent from HOST2.

The complement clear key values of both the exporter and importer are placed in the ICSF key output data set. Use these values as the clear key values for the CROSS statements on HOST2.

- At HOST2 where PCF/CUSP is installed, code the appropriate key generation utility program (KGUP) statement listed below for the environment in HOST2. When HOST2 has PCF:

```
CROSS name,KEYLOC=x,KEYREM=y,ADD
```

When HOST2 has CUSP:

```
CROSS name,KEYLOC=x,IKEYLOC=xx,KEYREM=y,IKEYREM=yy,ADD
```

Where:

name is the name of HOST2's CDRM

x is the key value from HOST1 of the IMPORTER key

y is the key value from HOST1 of the EXPORTER key

xx is the second half of double key value from HOST1's IMPORTER key

yy is the second half of double key value from HOST1's EXPORTER key

Figure 162 on page 671 illustrates the possible coding for cross-domain cryptographic sessions where HOST1 uses ICSF/MVS and HOST2 uses PCF/CUSP. APPL2A can initiate a cryptographic session with LU1A. APPL1A can initiate a cryptographic session with LU2A. APPL2A can initiate a cryptographic

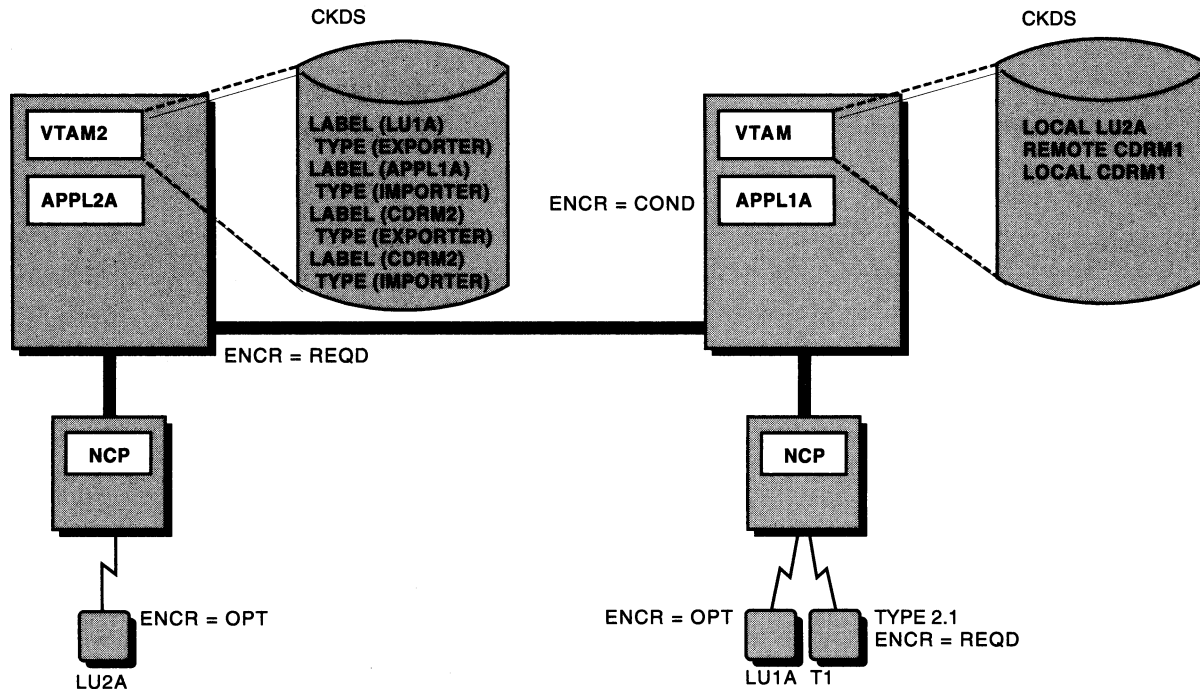


Figure 162. Cryptography in Multiple-Domain Environment (Hosts Use Different Cryptographic Services)

session with APPL1A. However, APPL1A cannot initiate a session with APPL2A because there is no TYPE(IMPORTER) statement coded in VTAM2 and APPL2A requires cryptography.

For the configuration in Figure 162 to have any encrypted sessions, start cryptography in both hosts before activating a session.

Changing a Logical Unit's Cryptographic Capability

If the VTAM encryption facility is installed, the cryptographic capability of the logical unit can be changed for future sessions by entering the MODIFY ENCR command. For the syntax of the command, see "MODIFY ENCR Command (MVS)" in *VTAM Operation*.

Appendix F. Command Lists: Dynamic Configuration of Channel-Attached Devices (MVS)

The IBM-supplied command lists are used to illustrate dynamic configuration of channel-attached devices (dynamic I/O). If you want to modify the command lists, this section describes how to call the command lists and what output the command lists return to assist you during modification. For general information, see “Dynamic Configuration of Channel-Attached Devices ” on page 147.

ISTDINFO: VTAM Device Information Services

The ISTDINFO command list accepts requests from a NetView REXX command list and uses the information to retrieve information about VTAM devices. ISTDINFO then returns the device information to the caller. You can call the ISTDINFO command list using following format:

```
ISTDINFO DEV_INFO, dev_string
```

The comma separating the parameters is mandatory. The parameters provide the following information:

DEV_INFO

Is the character string “DEV_INFO,” in either uppercase or lowercase.

dev_string

Is a character-string constant representing a hexadecimal device address. Device addresses consist of 1–4 hexadecimal characters (for example, X'F12').

Dependencies and Restrictions

If you are using the ISTDINFO command list, you should be aware of the following dependencies and restrictions:

- You can only IPL the configuration using an HCD I/O Definition (IODF). You cannot use MVS/CP to define your configuration. The SYS1.PARMLIB(LOADxx) member, which contains the IPL parameters, must have an IODF statement coded, not an MVS/CP statement.
- You need MVS Version 4 Release 2 or later releases.
- You can define only the following devices:
 - Local SNA clusters
 - Communications controllers.
- The devices must be defined to MVS using the IBM-supplied unit information modules.
- Complete device information is returned only when the hardware is self-describing.

Output Variable

The ISTDINFO command list returns information in the REXX special variable RESULT. RESULT is a string containing the following tokens:

- return_status
- device_address
- device_class
- self_describing
- device_type
- device_model
- defined_type
- generic_family
- PU_type
- TG_number
- IPL_online
- current_online
- allocate
- config_token
- maxbufru
- maxdata
- logical_terminals
- owner
- ncp_pu_type
- ncp_timeout
- ncp_delay
- ncp_maxbufru
- ncp_inbfrs
- ncp_maxdata
- ncp-transfr
- ncp_bfrpad
- ncp_TPS_feature.

Output Tokens

The following describes the information contained in the tokens:

return_status

the returned status from the device information services routine:

0	device specific information returned
2	invalid option string specified for parameter one
3	invalid device address specified for parameter two
4	no VTAM supported devices
5	unsupported operating system environment
101	internal error 101
102	internal error 102
201	internal error 201
301	internal error 301
302	internal error 302
303	internal error 303
304	internal error 304
305	internal error 305
306	internal error 306
307	internal error 307
308	internal error 308

309	internal error 309
310	internal error 310
311	internal error 311
401	internal error 401
501	internal error 501
502	internal error 502
503	internal error 503
504	internal error 504
505	internal error 505
601	internal error 601
602	internal error 602
603	internal error 603
604	internal error 604
605	internal error 605
606	internal error 606
607	internal error 607
608	internal error 608
609	internal error 609
610	internal error 610
611	internal error 611

device_address

address of the device associated with the returned information. This is not necessarily the same as the input *dev_string*.

device_class

device class stored in the unit control block (for example, COMM, DISP or CTC)

self_describing

an indication of whether the device is self-describing (YES or NO)

device_type

actual device type returned to MVS during VARY ONLINE processing, or *N/A* if not applicable

device_model

model number for this device, or *N/A* if not applicable

defined_type

defined device type specified by the user. For example, 3705 is specified for a 3705, 3725, 3745, or *N/A* if not applicable. This should always be defined.

generic_family

generic family of this device specified by the device: 3745 for 3745, or *N/A* if not applicable

PU_type

physical unit type of the device (for example, 2 or 21 or 4, or *N/A* if not applicable)

TG_number

transmission group number of the device: 0–255, or *N/A* if not applicable

IPL_online

status of the device at IPL: ONLINE, OFFLINE, or *N/A* if not applicable

current_online

status of the device currently: ONLINE or OFFLINE

allocate

status of allocation: ALLOCATED or UNALLOCATED

config_token

token representing the current I/O configuration. The actual I/O configuration is not printable in EBCDIC.

maxbufnu

maximum number of buffers to be set aside by the host, or *N/A* if not applicable

maxdata

maximum number of bytes the PU can receive in one PIU, or *N/A* if not applicable

logical_terminals

character string of 256 ones and zeroes. A one indicates a logical terminal exists at that offset, or *N/A* if not applicable. This string defines the number of logical units to be generated and the local address for each.

owner

value entered into HCD for the subsystem or access method using this device, or *N/A* if not applicable. Any embedded blanks in the owner string are replaced with the underscore (_) character.

ncp_putype

value entered into HCD for PU type, or *N/A* if not applicable

ncp_timeout

value entered into HCD for attention response timeout for the device, or *N/A* if not applicable

ncp_delay

value entered into HCD for maximum delay before attention to channel, or *N/A* if not applicable

ncp_maxbufnu

value entered into HCD for maximum number of host buffers allocated for the device, or *N/A* if not applicable

ncp_inbfrs

value entered into HCD for initial allocation of control buffers for the device, or *N/A* if not applicable

ncp_maxdata

value entered into HCD for the maximum number of bytes the PU can receive in one data transfer for the device, or *N/A* if not applicable

ncp_transfr

value entered into HCD for the number of NCP buffers for MAXDATA transfer for the device, or *N/A* if not applicable

ncp_bfrpad

value entered into HCD for number of pad characters to the access method for the device, or *N/A* if not applicable

ncp_TPS_feature

value entered into HCD for mode of the two-processor switch feature for the device: TPS, TCS, NONE, or *N/A* if not applicable.

ISTDEFIN: VTAM Device Information Services

The ISTDEFIN command list accepts one or more device addresses, gets information about each device, defines it to VTAM, ensures that each device is online, and activates the VTAM resource definitions. You can call the ISTDEFIN command list using the following format:

Call ISTDEFIN *func_code dev_str*

The parameters provide the following information:

func_code

func_code is one of the following function codes that indicate how the input device addresses correlate to the device processed by the command list:

- | | |
|---|--|
| 1 | Process only device addresses passed to this command list. |
| 2 | Use input device address as a base address to locate communications, display or CTC device to process. |

dev_str

dev_str is a string of one or more hexadecimal device addresses separated by blanks.

For example, if you wanted to call the ISTDEFIN command list with a function code of 1 and three device addresses X'00A', X'123', and X'FE3', you would send the following:

```
ISTDEFIN 1 A 123 FE3
```

Dependencies and Restrictions

If you are using the ISTDEFIN command list, you should be aware of the following dependencies and restrictions:

- You need NetView 1.3 or later releases or a product providing a similar interface and services.
- ISTDEFIN must execute in the NetView address space.
- You need MVS Version 4 Release 2 or later releases.
- You can define only the following devices:
 - Local SNA clusters
 - Communications controllers.
- You need to concatenate USER1.AUTO.VTAMLST with SYS1.VTAMLST.
- You can only define NCP major nodes if they are channel-attached.
- The IBM-supplied naming convention supports only 36 subarea addresses.
- Error messages for the VARY NET,ACT command are diagnosed only to the extent that syntax errors in the resource definitions generated by this sample are identified.

Output Variable

The ISTDDEFIN command list returns information in the REXX special variable RESULT. RESULT is a string containing the following tokens:

```
return_status
device_address1
device_rc1
device_address2
device_rc2
:
device_addressn
device_rcn
```

where n is the number of devices processed by the command list.

Output Tokens

The following describes the information contained in the tokens:

return_status

highest return code encountered while processing the device string

device_address_i

actual device address processed

device_rc_i

return code that resulted from processing device_address_i. The possible values are:

0	device successfully defined and activated
1	device address not found
2	invalid option string (parameter 1) specified on ISTDINFO call
3	invalid device address (parameter 2) specified on ISTDINFO call
4	device is currently allocated
5	operating environment not supported
6	operator cancelled processing for device
7	timer expired while varying device online
8	error occurred while updating USER1.AUTO.VTAMLST
9	VARY ONLINE command failed for device
10	PU type not supported by VTAM
11	timer expired during VARY ACTIVATE
12	error activating device
14	device not defined to come online at IPL
15	device does not support sysdef automation
101	ISTDINFO internal error 101
102	ISTDINFO internal error 102
201	ISTDINFO internal error 201
301	ISTDINFO internal error 301
302	ISTDINFO internal error 302
303	ISTDINFO internal error 303
304	ISTDINFO internal error 304
305	ISTDINFO internal error 305
306	ISTDINFO internal error 306
307	ISTDINFO internal error 307
308	ISTDINFO internal error 308
309	ISTDINFO internal error 309

310	ISTDINFO internal error 310
311	ISTDINFO internal error 311
401	ISTDINFO internal error 401
501	ISTDINFO internal error 501
502	ISTDINFO internal error 502
503	ISTDINFO internal error 503
504	ISTDINFO internal error 504
505	ISTDINFO internal error 505
601	ISTDINFO internal error 601
602	ISTDINFO internal error 602
603	ISTDINFO internal error 603
604	ISTDINFO internal error 604
605	ISTDINFO internal error 605
606	ISTDINFO internal error 606
607	ISTDINFO internal error 607
608	ISTDINFO internal error 608
609	ISTDINFO internal error 609
610	ISTDINFO internal error 610
611	ISTDINFO internal error 611.

Appendix G. Message Translation Using the MVS Message Service

National language support uses the MVS Message Service (MMS) for the translation of VTAM messages. This section describes how MMS is used to translate VTAM messages. For an overview of national language support, see “National Language Support” on page 157. The following topics are discussed in this section:

- Overview of MMS support
- Internal translation
- External translation
- Skeleton file use.

Overview of MMS Support

The MVS Message Service translates VTAM messages using message skeleton files for particular languages. MMS finds a skeleton file matching the language requested and then searches for the message and returns the text from the skeleton file.

VTAM provides a base US English message skeleton file named ISTVTMEU. This file contains all VTAM messages including USS, TSO/VTAM, and logon manager messages. The way in which ISTVTMEU is used depends on whether the translation of the message occurs internally or externally to VTAM.

Internal Translation

VTAM uses internal translation to translate USS and TSO/VTAM end-user messages. These messages must be translated internal to VTAM so that they can be sent on either a SSCP-LU (for USS) or LU-LU session (for TSO/VTAM). For end-user messages, ISTVTMEU defines the US English text that should be used for these messages. This text can be changed, providing a function similar to USS tables.

Selecting Internal Translation

To use internal translation:

1. To support a language other than US English, first define a language skeleton file to MMS. The language skeleton file must have the language code or language name that will be specified by the logical unit. If the required language skeleton file does not exist when the USS command is entered and the MMS is active then the USS command will be rejected. For TSO/VTAM, default US English messages are issued if the required language skeleton file does not exist.
2. An LU must explicitly request that translation be performed before messages will be translated internally.
 - For USS, the LANG operand on USS commands determines the language.

- For TSO/VTAM, the language can be determined from the LANG or LANGTAB operands (the language information is provided on the CINIT) or by the TSO PROFILE command with the PLANG operand.

Defining Messages for Internal Translation

The IBM-supplied file, ISTVTMEU, contains messages for internal and external translation. The end-user messages defined in ISTVTMEU for internal translation are USSMSG01 through USSMSG13 for USS and IKT00201I through IKT00405I for TSO/VTAM. For example, USS message 1 is defined as follows:

```
USSMSG01          INVALID &1. COMMAND SYNTAX      &TIME.
```

You can change the wording of this message to a different wording if desired. The &1. is an example of a substitution token and is used here to identify where the incorrect command name is to be placed in the message. Additional substitution tokens are &2., &3. and so on. Also, the &TIME., &DATE., &LUNAME., and &NETID. substitution tokens are used for the time, date, LU name, and network ID. If you want to provide USS messages in other languages using MMS, another skeleton file can be created for the desired language with the desired message text. For example, you could code the following in a new file named ISTVTFRA with the text in French:

```
USSMSG01          THE SYNTAX FOR COMMAND &1. IS INVALID
```

Then, if the end user specified LANG(FRA) on a USS command the above message text would be provided in the case of a USS command syntax error. The above message could also be defined as a multi-line message and when translation occurs, all of the lines in the translated message are written to the LU. There is a 23-line limit on the number of lines in a translated message for USS and an 8-line limit for TSO/VTAM.

USS messages defined with MMS skeleton files must follow the same restrictions as USS messages defined with USS tables. Only a limited character set is available to SNA logical units on the SSCP-LU session.

For TSO/VTAM messages, any character set supported by the end-user terminal can be used including double byte character sets. Using incompatible character sets and code pages can result in erroneous output or I/O errors.

External Translation

External translation occurs outside of VTAM and is only available to authorized TSO/E 2.2 users using multiple console support (MCS) extended consoles.

Selecting External Translation

In external translation, the language for the extended console is determined by the user's profile, which is set with the PLANG operand of the TSO PROFILE command.

When a language other than US English is selected, the following occurs:

1. ISTVTMEU is used to parse the US English message to determine the variable text.
2. MMS looks up the corresponding translated message and substitutes the variable text.

ISTVTMEU and ISTINCNO must match for external translation to work correctly. It is recommended that you use the IBM-supplied copies of these files and that you do not use the USSTAB start option. Customization of messages can then be accomplished by changing the language skeleton files used by MMS.

Defining Messages for External Translation

Messages for external translation are also defined in ISTVTMEU. All of the messages for external translation are messages that are written to the system console. You should not change the US English form of these messages in ISTVTMEU, because ISTVTMEU must correspond exactly with ISTINCNO for external translation to occur. For operator messages, you should use the IBM-supplied default table, ISTINCNO, rather than using a customized operator USS table.

For translation to other languages, provide a language skeleton file similar to ISTVTMEU with the translated text. For example, VTAM message IST001I is defined as follows in ISTVTMEU:

```
IST001I      001      VTAM START REJECTED - &1.
IST001I      002      &modnm. VTAM START REJECTED - &1.
```

The text of this message could be changed to French in a new skeleton file named ISTVTFRA. It is necessary to have two formats of this message: one for the message without the module name and one for the message with the module name. MMS will select the format that most closely matches the message received and perform the translation by selecting the corresponding message with the corresponding format number from the new French skeleton file.

For example, assume ISTVTFRA has the following definition for IST001I:

```
IST001I      001      &1. - REJECT START
IST001I      002      &modnm. &1. - REJECT START
```

With the above skeletons coded, a TSO/E 2.2 user using an MCS extended console with the primary language of FRA will have the following VTAM message translated as shown below:

```
Message issued by VTAM:
IST001I VTAM START REJECTED - TERMINATION IN PROGRESS
```

```
Translated message on MCS Extended console
IST001I TERMINATION IN PROGRESS - REJECT START
```

The translated messages can be in any language supported by MMS including double-byte character set languages. The only restriction is that the terminal being used for a given language must support that language with the correct character set and code page or erroneous output or I/O errors can occur. Translated messages can also be defined as multi-line messages if more than one line is needed to express the error in the translation language.

Skeleton File Use

ISTVTMEU, and any other translation language skeletons required by the installation, must be compiled by the MMS compiler before use. Also, MMS has to be active before VTAM can use any of the skeleton files.

ISTVTMEU is shipped in library SYS1.SISTDAT2. This member can be copied to another partitioned data set where other MMS skeleton files are kept. Skeleton files for languages other than US English must be provided by the installation. For information on compiling message skeleton files, see *MVS/ESA Planning: Operations*. For information on defining message skeleton files, see *MVS/ESA Application Development Guide: Assembler Language Programs*.

Appendix H. Forcing an APPN Route in a VTAM Network

This appendix describes how to force the path of a route through your network. You might want to force a route if you want certain sessions to use a specific path rather than the default path.

If you use only the defaults, the nodes and links are considered equal and the route with the fewest nodes and links will have the least weight. The route with the least weight is selected as the route for the session.

For example, if all the defaults are used in the network shown in Figure 163 and an LU in NN1 requests a session with an LU in NN3, the route chosen for the session will go from NN1 through NN2 to NN3.

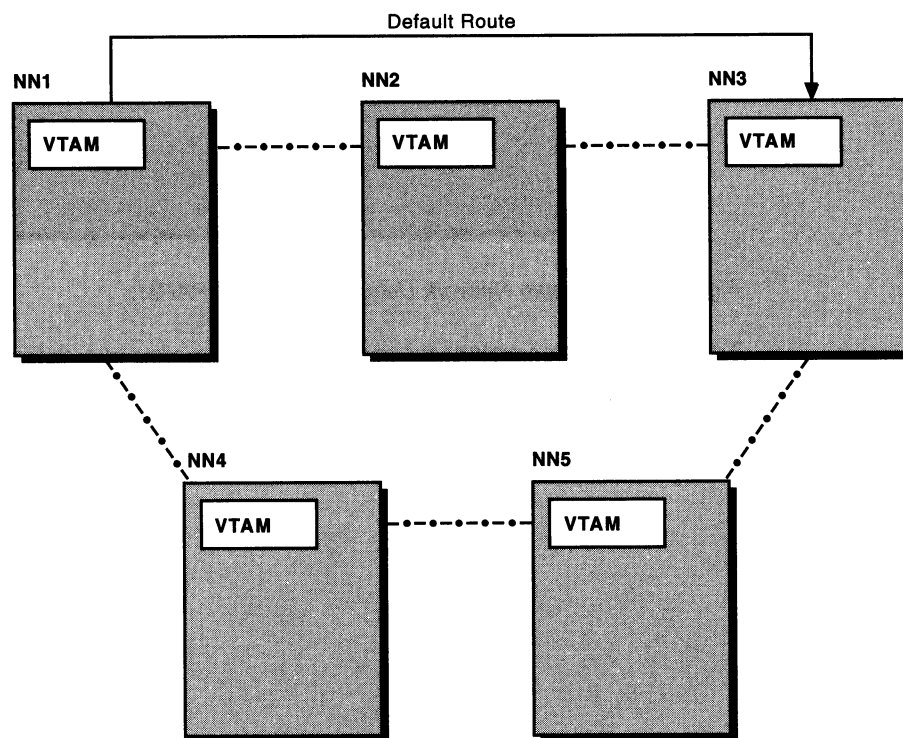


Figure 163. Sample Network Showing Default Route

If you want to force a different route for a certain type of session, you need to cause that route to have a lower weight than the original default route. For example, you might want the session between the LU in NN1 and the LU in NN3 in Figure 163 to be routed through NN4 and NN5. By defining a new APPN class of service and a transmission group (TG) profile, you can create a lower weight for the route you chose than the weight of the default route.

This example demonstrates using the UPARM1 operand to force a route to traverse NN1, NN4, NN5, and NN3. With the defaults, the values for UPARM1 for the links in the network would look like the example in Figure 164 on page 686.

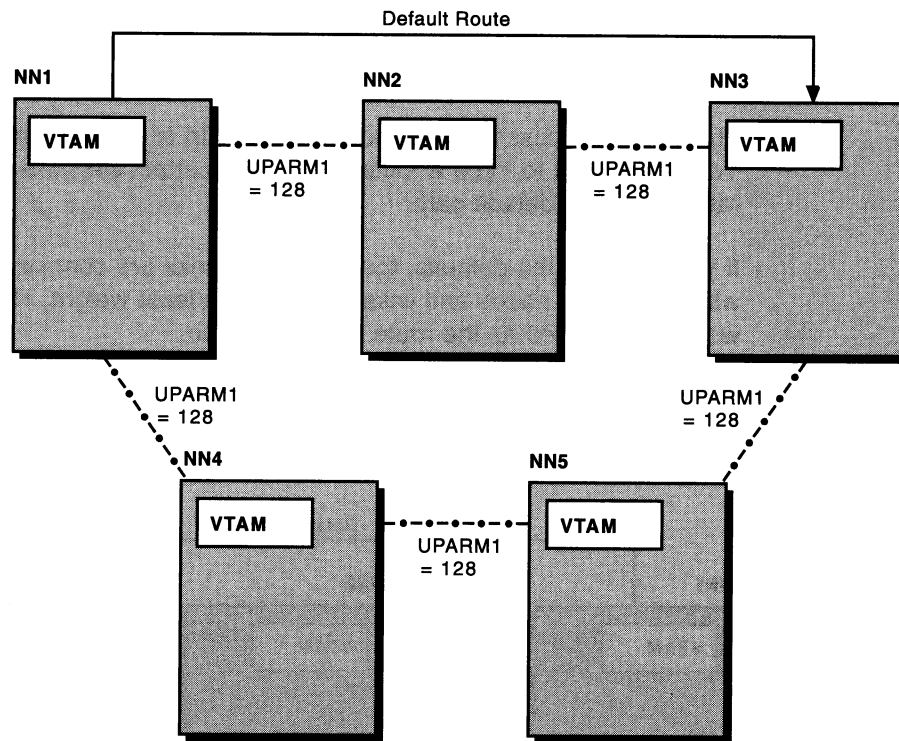


Figure 164. Sample Network Using Default for UPARM1

To create a lower weight for the chosen route, you can define a TG profile with a UPARM1 value of 0 and specify that TG profile on the PUs associated with the links between NN1 and NN4, NN4 and NN5, and NN5 and NN3. The UPARM1 values in the network would then appear as they do in Figure 165 on page 687.

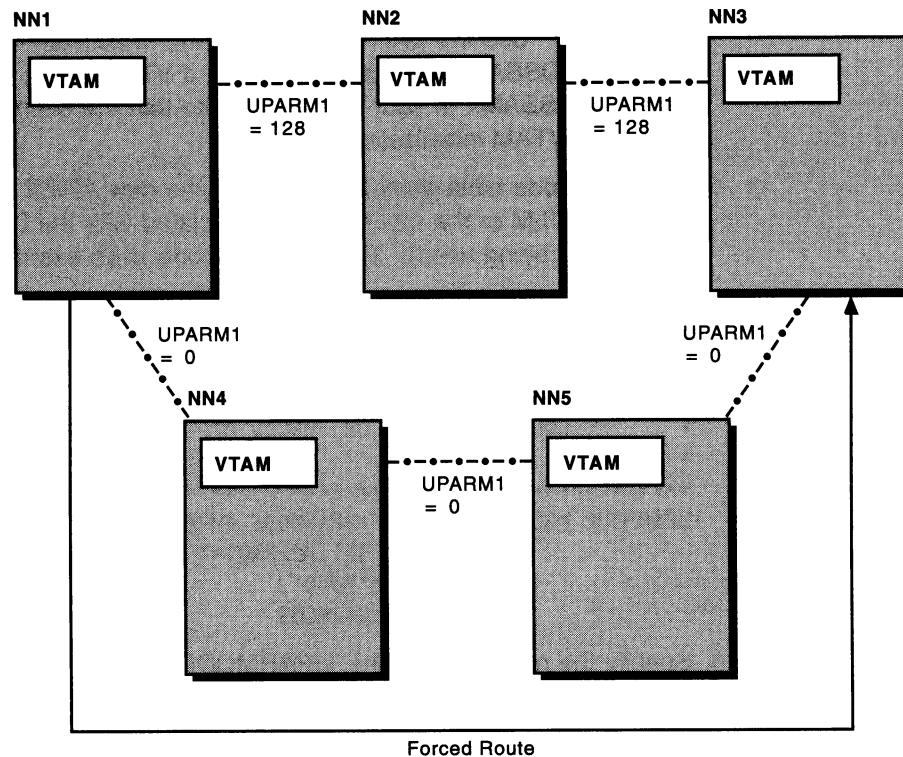


Figure 165. Sample Network Using TG Profile on Some Links

By defining UPARM1 on the TG profile as 0, the links that specify that TG profile have a unique characteristic. You can define a new class of service that will give preference to links with a lower UPARM1 value. The following steps describe how to define the TG profile and define a class of service to take advantage of the UPARM1 value:

1. Code a TG profile with UPARM1=0 as shown in the following example.

```

*****
* TG Profile to change UPARM1 value                                     *
*****
*
UPARMLOW TGP    UPARM1=0
  
```

2. Specify this TG profile for the PUs associated with each line in the preferred route.
3. Code a class of service that gives preference to links using the TG profile.
 - a. Copy the eighth LINEROW and NODEROW statements from the IBM-supplied COS definition #CONNECT (found in the COSAPPN member of SYS1.VTAMLST).
 - b. Make a second copy of the LINEROW statement.
 - c. On the LINEROW statements, modify only the NUMBER, UPARM1, and WEIGHT operands.
 Modify the first LINEROW statement to specify NUMBER=1, UPARM1=(0,99), and WEIGHT=30.
 Modify the second LINEROW statement to specify NUMBER=2, UPARM1=(100,255), and WEIGHT=240.

4. File this new APPN COS definition in SYS1.VTAMLST under the new member, COSSAMP. Use the VARY ACT command to activate this definition by specifying ID=COSSAMP. (As an alternative, if VTAM is not already active, you can specify COSSAMP in your CONFIG list so that the definition will be activated as part of VTAM initialization.)
5. Define a mode table entry that specifies the new APPN COS definition and add it to ISTINCLM or the mode tables associated with the LUs (if the default mode table is not being used). The following code is an example of the mode table entry.

```

                TITLE 'UPARMODE'
*****
*
*          LOGMODE ENTRY FOR FORCED APPN ROUTE EXAMPLE -
*
*****
UPARMODE MODEENT LOGMODE=UPARMODE,FMPROF=X'03',
                TSPROF=X'03',PRIPROT=X'B1',SECPROT=X'A0',
                COMPROT=X'3040',
                APPNCOS=UPARMCOS

```

6. Specify the new mode name, UPARMODE, when requesting the session.
7. Because UPARMCOS is the APPN COS definition used by the route selection process, the NN1 to NN2 to NN3 route has a weight of $240 + 160 + 240 = 640$ based on the values of the WEIGHT operand from the UPARMCOS definition. Because the NN1 to NN4 to NN5 to NN3 route has a weight of $30 + 160 + 30 + 160 + 30 = 410$ based on the values of the WEIGHT operand, it will be the chosen least-weight route.

Note: For this example, it is required that the new mode and class-of-service tables be stored in the nodes that are the end points of the session path; however, we strongly recommend that the class of service and mode tables be consistent among all the nodes in the network.

The following example shows the new user-defined class of service.

```
*****
* user-defined class-of-service definition *
*****
UPARMCOS APPNCOS PRIORITY=MEDIUM      transmission priority
      LINEROW WEIGHT=30,                TG weight *
      NUMBER=1,                        line row number *
      UPARM1=(0,99),                   user defined parameter 1 *
      UPARM2=(0,255),                  user defined parameter 2 *
      UPARM3=(0,255),                  user defined parameter 3 *
      CAPACITY=(MINIMUM,MAXIMUM),      line speed *
      COSTTIME=(0,255),                 cost per connect time *
      COSTBYTE=(0,255),                 cost per byte transmitted *
      PDELAY=(MINIMUM,MAXIMUM),         propagation delay *
      SECURITY=(UNSECURE,MAXIMUM)       security level for TG
      LINEROW WEIGHT=240,              TG weight *
      NUMBER=2,                        line row number *
      UPARM1=(100,255),                 user defined parameter 1 *
      UPARM2=(0,255),                  user defined parameter 2 *
      UPARM3=(0,255),                  user defined parameter 3 *
      CAPACITY=(MINIMUM,MAXIMUM),      line speed *
      COSTTIME=(0,255),                 cost per connect time *
      COSTBYTE=(0,255),                 cost per byte transmitted *
      PDELAY=(MINIMUM,MAXIMUM),         propagation delay *
      SECURITY=(UNSECURE,MAXIMUM)       security level for TG
      NODEROW NUMBER=2,                node row number *
      WEIGHT=160,                      node weight *
      CONGEST=(LOW,HIGH),               congestion *
      ROUTERES=(0,255)                 route addition resistance
```


Appendix I. Border Node Connection Types

This appendix lists connection types for various VTAM and partner node combinations. For further discussion of border nodes and APPN multiple network connectivity support, see "APPN Multiple Network Connectivity" on page 545.

Table 43 (Page 1 of 4). Connection Type for Selected VTAM and Partner Node Combinations

Partner Node Characteristics	Local VTAM Characteristics							Connection Type
	BN		NATIVE			NETID		
	YES	NO or Not Coded	YES	NO	Not Coded	Same	Different	
Network node	✓		✓			✓		Native connection, CP-CP sessions supported
Network node	✓		✓				✓	Native connection, CP-CP sessions not supported
Network node	✓			✓		✓		Peripheral connection, CP-CP sessions supported
Network node	✓			✓			✓	Peripheral connection, CP-CP sessions supported
Network node	✓				✓	✓		Native connection, CP-CP sessions supported
Network node	✓				✓		✓	Peripheral connection, CP-CP sessions supported
Network node		✓			✓	✓		Native connection, CP-CP sessions supported
Network node		✓			✓		✓	Native connection, CP-CP sessions not supported
Peripheral border node	✓		✓			✓		Native connection, CP-CP sessions supported
Peripheral border node	✓		✓				✓	Connection failed by local VTAM
Peripheral border node	✓			✓		✓		Peripheral connection, CP-CP sessions supported
Peripheral border node	✓			✓			✓	Peripheral connection, CP-CP sessions supported

Table 43 (Page 2 of 4). Connection Type for Selected VTAM and Partner Node Combinations

Partner Node Char- acteristics	Local VTAM Characteristics							Connection Type
	BN		NATIVE		NETID			
	YES	NO or Not Coded	YES	NO	Not Coded	Same	Dif- ferent	
Peripheral border node	✓				✓	✓		Native connection, CP-CP sessions sup- ported
Peripheral border node	✓				✓		✓	Peripheral con- nection, CP-CP ses- sions supported
Peripheral border node		✓			✓	✓		Native connection, CP-CP sessions sup- ported
Peripheral border node		✓			✓		✓	Peripheral con- nection (managed by partner node), CP-CP sessions sup- ported
Extended border node, NATIVE=YES	✓		✓			✓		Native connection, CP-CP sessions sup- ported
Extended border node, NATIVE=YES	✓		✓				✓	Native connection, CP-CP sessions not supported
Extended border node, NATIVE=YES	✓			✓		✓		Connection failed by partner node
Extended border node, NATIVE=YES	✓			✓			✓	Connection failed by partner node
Extended border node, NATIVE=YES	✓				✓	✓		Native connection, CP-CP sessions sup- ported
Extended border node, NATIVE=YES	✓				✓		✓	Connection failed by partner node
Extended border node, NATIVE=YES		✓			✓	✓		Native connection, CP-CP sessions sup- ported
Extended border node, NATIVE=YES		✓			✓		✓	Native connection, CP-CP sessions not supported
Extended border node, NATIVE=NO	✓		✓			✓		Connection failed by local VTAM
Extended border node, NATIVE=NO	✓		✓				✓	Connection failed by local VTAM
Extended border node, NATIVE=NO	✓			✓		✓		Extended connection, CP-CP sessions sup- ported

Table 43 (Page 3 of 4). Connection Type for Selected VTAM and Partner Node Combinations

Partner Node Char- acteristics	Local VTAM Characteristics							Connection Type
	BN		NATIVE			NETID		
	YES	NO or Not Coded	YES	NO	Not Coded	Same	Dif- ferent	
Extended border node, NATIVE=NO	✓			✓			✓	Extended connection, CP-CP sessions sup- ported
Extended border node, NATIVE=NO	✓				✓	✓		Extended connection, CP-CP sessions sup- ported
Extended border node, NATIVE=NO	✓				✓		✓	Extended connection, CP-CP sessions sup- ported
Extended border node, NATIVE=NO		✓			✓	✓		Peripheral con- nection (managed by partner node), CP-CP sessions sup- ported
Extended border node, NATIVE=NO		✓			✓		✓	Peripheral con- nection (managed by partner node), CP-CP sessions sup- ported
Extended border node, NATIVE not coded	✓		✓			✓		Native connection, CP-CP sessions sup- ported
Extended border node, NATIVE not coded	✓		✓				✓	Connection failed by partner node
Extended border node, NATIVE not coded	✓			✓		✓		Extended connection, CP-CP sessions sup- ported
Extended border node, NATIVE not coded	✓			✓			✓	Extended connection, CP-CP sessions sup- ported
Extended border node, NATIVE not coded	✓				✓	✓		Native connection, CP-CP sessions sup- ported
Extended border node, NATIVE not coded	✓				✓		✓	Extended connection, CP-CP sessions sup- ported
Extended border node, NATIVE not coded		✓			✓	✓		Native connection, CP-CP sessions sup- ported
Extended border node, NATIVE not coded		✓			✓		✓	Peripheral con- nection (managed by partner node), CP-CP sessions sup- ported

Border Node Connection Types

Table 43 (Page 4 of 4). Connection Type for Selected VTAM and Partner Node Combinations

Partner Node Characteristics	Local VTAM Characteristics							Connection Type
	BN		NATIVE			NETID		
	YES	NO or Not Coded	YES	NO	Not Coded	Same	Different	
Notes:								
1. The local VTAM Node is always configured with the NODETYPE=NN specified.								
2. XNETALS=YES is assumed whenever the NETIDs of the two nodes are different. If XNETALS=NO is specified, then connections between two nodes with different NETIDs fail.								
3. When BN=YES is coded, XNETALS=YES is the default start option. When BN=YES is not coded, XNETALS=NO is the default start option.								

Appendix J. VTAM Restricted Materials

The VTAM modules and macroinstructions listed in this appendix are considered to be Restricted Materials of IBM and are available through the View Program Listings (VPL) application. This information can be accessed by using ServiceLink or Dial IBM. All other VTAM modules are designated as object code only (OCO).

Table 44. Accounting Purpose Modules

Module	Description	MVS	VM	VSE
ISTINM01	Initializes VTAM environment and gets storage for ATCVT	Y		
ISTINV01	Initializes VTAM environment and gets storage for ATCVT		Y	
ISTINCHA	Drives the VTAM initialization process			Y
ISTAPCFS	When VTAM terminates, frees any common storage allocated to VTAM	Y	Y	
ISTDVCRC	Inserts entries into the DVT string after each entry for ISTTSCCS and before each entry of ISTTSCCR, to obtain information from the PIU	Y	Y	Y
ISTACCSA	Dial connect	Y	Y	Y
ISTDECSA	Dial disconnect	Y	Y	Y
ISTACCR6	Link activation	Y	Y	Y
ISTACCI3	Link activation request	Y	Y	Y
ISTRACLN	Line trace	Y	Y	Y
ISTDECK2	Link deactivation	Y	Y	Y
ISTINCTR	Trace initialization	Y	Y	Y
ISTTSCNI	Process network services requests	Y	Y	Y
ISTVTLOD	VTAM load list	Y	Y	Y
ISTRACAP	API trace (internal trace)	Y		Y
ISTRACBF	Buffer trace (external trace)	Y	Y	Y

Table 45. User Replaceable or Modifiable Modules

Module	Description	MVS	VM	VSE
ISTPUCWC	Modifies the VR window size	Y	Y	Y
ISTMGC01	CNM routing table (IBM-provided default table)	Y	Y	Y
ISTINCLM	Logon mode table	Y	Y	Y
ISTINCDT	Session-level USS table	Y	Y	Y
ISTINFIC	Initializes the ATCVT and configuration table (ISTCONFT)	Y	Y	Y

Table 46. VTAM Message Modules

Module	Description	MVS	VM	VSE
ISTCFCMM	English text of VTAM messages	Y	Y	Y
ISTINCNO	Formatted English text of VTAM messages	Y	Y	Y
ISTCFCML	Defines variables and variable lengths for VTAM messages	Y	Y	Y

VSE The following link scripts are available on microfiche.

ISTLKABV	ISTLKDRT	ISTLKOSM	ISTLKSVT
ISTLKART	ISTLKDSR	ISTLKPDC	ISTLKTPS
ISTLKATI	ISTLKE00	ISTLKPRT	ISTLKTP2
ISTLKATR	ISTLKE01	ISTLKPR2	ISTLKTRS
ISTLKA00	ISTLKHLM	ISTLKP00	ISTLKTO0
ISTLKA25	ISTLKIBX	ISTLKP01	ISTLKTO9V
ISTLKBRT	ISTLKIEC	ISTLKRNT	ISTLKUDL
ISTLKCHC	ISTLKIEF	ISTLKRRT	ISTLKULS
ISTLKCLU	ISTLKIIR	ISTLKRTM	ISTLKURT
ISTLKCR2	ISTLKLNF	ISTLKSCC	ISTLKUTS
ISTLKCSC	ISTLKLNP	ISTLKSCT	ISTLKUVL
ISTLKCSM	ISTLKMRT	ISTLKSDN	ISTLK001
ISTLKCTC	ISTLKMST	ISTLKSFV	ISTLK002
ISTLKCUM	ISTLKNDP	ISTLKSHO	ISTLK003
ISTLKC00	ISTLKNF0	ISTLKSJC	ISTLK004
ISTLKC01	ISTLKNLR	ISTLKSJC	ISTLK005
ISTLKC5C	ISTLKNRO	ISTLKSLS	ISTLK007
ISTLKC6C	ISTLKNRT	ISTLKSRT	ISTLK008
ISTLKDLM	ISTLKNUT	ISTLKSSC	ISTLK25F
ISTLKDLS	ISTLKNVT	ISTLKSSL	ISTLK25P
ISTLKDPC	ISTLKNZ1	ISTLKSVL	

Appendix K. VTAM AnyNet Feature (MVS)

The AnyNet family of products is a group of workstation and host-based functions that implement the Multiprotocol Transport Networking (MPTN) architecture. MPTN architecture supports:

- Mixed protocol networking: running application programs that were designed to operate over one transport protocol—such as SNA or TCP/IP—over additional transport protocols.
- Network interconnection: connecting matching application program partners across networks of different types, for example, allowing a socket application running on TCP/IP network to communicate with a socket application running on an SNA network.

The VTAM AnyNet feature enables application programs to communicate, without change, over different transport networks and across interconnected networks. With VTAM's AnyNet functions, you can reduce the number of transport networks, thus reducing operational complexity. These benefits are gained without modification to existing application programs or hardware.

Overview of Multiprotocol Transport Networking (MPTN)

In March, 1993, IBM announced Multiprotocol Transport Networking, (MPTN), an architecture that makes fully integrated communication among mixed protocol networks possible. MPTN architecture is a key part of the common transport semantics (CTS) layer. MPTN expands end user application choice and promotes network simplification of the underlying transport network.

Peer and Transport Relationships

Figure 166 shows that there are two sets of relationships that need to be defined: peer relationships and relationships between transport users and transport providers. The term *matching* describes a horizontal relationship between peers.

In the MPTN architecture, transport users must always match. “Matching” means that the transport users have the same application support and belong to the same application programming family. For example, two sockets programs match or two Common Programming Interface for Communications (CPI-C) application programs match, but a sockets application program and a CPI-C application program do not match.

The terms *native* and *nonnative* describe a vertical relationship between a user and its corresponding provider. A transport network that provides the address type and transport characteristics assumed in the transport user's design is native to that transport user, such as APPC over SNA. A transport provider that does not provide that particular transport user address type and those transport characteristics is nonnative, such as Sockets over SNA.

Application programs that are designed to assume a particular transport provider are native to that transport provider. At the same time, they are nonnative to another transport provider. A *native node* is a node with no MPTN capability. For example, a node with APPC applications, running only over an SNA transport, is a native node.

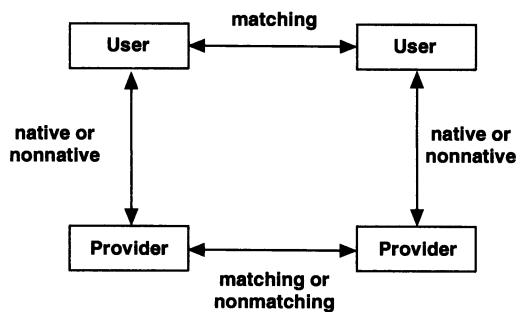


Figure 166. Illustration of the Use of Matching and Native Relationships

MPTN Architecture Functions

The MPTN components, working together, provide the following functions:

- Compensation
- Address mapping
- Network interconnection
- Network management

Compensation

A compensation function bridges the gap between the needs of the transport user and the services provided by the underlying transport provider. For example, the transport user may require maintaining record boundaries over a transport provider that does not maintain record boundaries. A common set of compensations exists that covers most of the mismatched functions between transport users and transport providers.

Address Mapping

An MPTN connection request must include an identifier of the partner transport user. Different transport providers use different address formats to identify the location of programs or users in the network. The MPTN architecture provides various techniques for mapping from the transport addresses provided by the transport users to those required by the corresponding transport providers.

The goal of MPTN address mapping services is to allow any higher-level component or application program using any transport address structure to run over any transport provider without requiring either to change the transport address structure that it uses today.

Network Interconnection

Figure 167 on page 699 illustrates how MPTN transport gateways are used to interconnect different transport providers when two matching application programs are running over dissimilar transport providers.

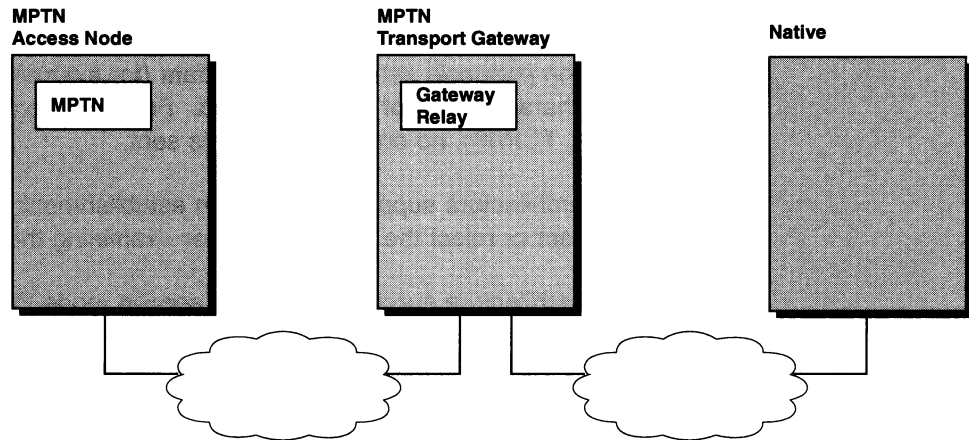


Figure 167. MPTN Gateway

Network Management

Management of an MPTN network requires correlation among signals produced by different native management protocols as a result of incidents in the transport networks. Network management also involves correlating these problems to transport users that are being coordinated by a different set of management tools.

MPTN Transport Services

By making a general set of transport services available, the MPTN architecture enables higher-layer application support services, such as remote procedure call (RPC), to be written without any specific transport protocol in mind. The set of MPTN transport services was originally derived from the services provided by four common transport providers: NetBIOS, OSI layer 4, SNA, and TCP/IP. Because these four protocol families provide a broad class of transport services, the MPTN architecture is expected to be applicable to other providers as well.

There are two approaches to deciding what transport services to provide to users:

- One approach is to select only the services that occur in all the transport technologies being considered. Such an interface has minimal function and requires you to add much function that is already provided in many protocols.
- Another approach, and the one chosen for the MPTN architecture, is to select all generally useful and necessary services, and to provide you with some general compensations for missing functions.

Transport services described in this chapter fall into two broad classes: *connection-oriented* and *connectionless*.

Connection-Oriented Services

Connection-oriented services establish a logical connection between two partners for the duration of the period they communicate with each other. The connection provides reliable, in-order (first-in, first-out) delivery of data. There are three distinct phases in a connection-oriented data exchange:

- Connection establishment
- Data transfer
- Connection termination

Connection Establishment: The logical connection between two partners is established during a connection-establishment phase. The transport users of some communication protocols send connection data (for example, a BIND in SNA) to specify the characteristics of the connection. For other communication protocols (for example, TCP/IP), no connection data is sent.

The MPTN architecture supports connection establishment with or without data. You can accept or reject the connection after examining the connection data.

The MPTN architecture supports a general service mode that can map to SNA's class of service, OSI's quality of service, and TCP/IP's type of service, so you can control the quality of service over one or more nonnative transport providers.

Data Transfer: Reliable data transfer is provided by the transport provider using MPTN. When an unreliable data link or network layer protocol is used, the reliability is assured by the transport provider, such as SNA or TCP. Other characteristics of the data transfer phase include whether a stream or record model is used and whether expedited data is supported.

Some user applications and higher-layer protocols assume data record boundaries are preserved by the underlying transport provider. Others send data as a byte stream with no record boundary delineation.

Both data-delivery models are supported by the MPTN architecture. For record models, the maximum record sizes can vary among the various transport providers. The MPTN architecture allows the transport user to select the maximum record size based on what the user expects from its native transport provider.

Some user applications and other higher-layer components send expedited data.

The MPTN architecture assures that expedited data:

- Is not subject to the same data flow control as normal data
- Is not delivered later than any normal data sent after it, but can be delivered ahead of normal data sent before it
- Can be delivered to the transport user even if the user is not receiving normal data
- Can be distinguished from normal data when sent or received

Connection Termination: During connection termination, some transport users expect to be able to deliver connection termination data to the partner, such as the reason for the termination. Other transport users do not require this ability. Some transport users expect to terminate one direction of the connection at a time (simplex termination), while others expect to terminate the entire connection at once (duplex termination). Similarly, some transport users expect the connection termination data to stay inline behind data sent earlier (orderly termination), while others permit it to overtake data sent earlier (abortive termination). All of these connection termination semantics are provided by the MPTN architecture.

Connectionless Services

Connectionless service is the mode of data transfer in which different units of data are passed independently through the network. Delivery is provided on a best-effort basis, but undetected packet loss can occur if, for example, the network becomes congested. There is no guarantee that data units will be delivered in the same sequence that the sender issued them or that duplicates will not be delivered.

The basic unit of connectionless data transfer is the *datagram*, which is a data packet that carries information sufficient for routing from a source to its intended destination.

Different transport providers have different maximum datagram size limitations. MPTN architecture supports the maximum datagram size expected by the transport user.

Connectionless modes supported by MPTN architecture are:

- Unicast datagrams
- Multicast datagrams
- Broadcast datagrams

Unicast Datagrams: The MPTN architecture allows a transport user to send a datagram to a single partner. This is referred to as a unicast datagram.

Multicast Datagrams: The MPTN architecture allows a transport user to send a datagram to a group address. This is referred to as a multicast datagram. The sender need not know which programs belong to the group and need not be a member of the group itself.

Since multicast service is a significant service that is provided on various local area networks, it is included in MPTN architecture. MPTN architecture support for multicast requires that all members of a multicast group have the same address and be located in a common network. This is referred to as a multicast datagram.

Broadcast Datagrams: Broadcast datagram service is treated as a special case of multicast datagrams service. A broadcast datagram is a datagram sent to all transport users in a given network that have indicated a willingness to receive broadcast messages. Transport users that expect to use a broadcast function join this broadcast group. MPTN architecture limits broadcast distribution to only the members of a single network, not all interconnected networks.

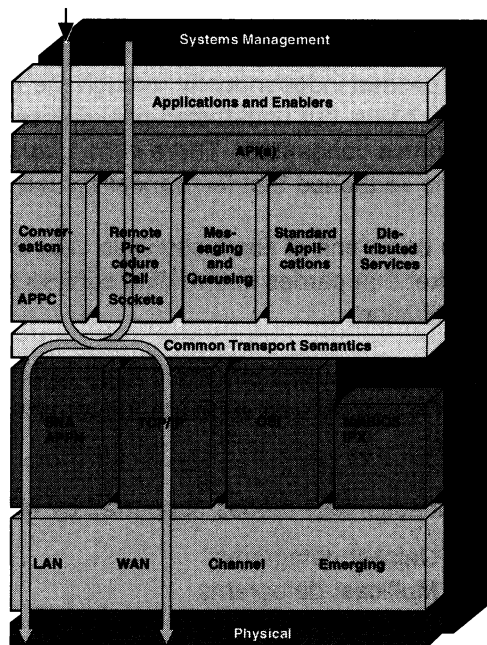


Figure 168. Mixed Protocol Operation

MPTN implementations will be announced and delivered under the AnyNet family product name. Figure 168 illustrates a mixed protocol operation. You can expect to see AnyNet products delivered on different multiprotocol platform combinations.

Where to Use MPTN Connections

If you have many networking protocols in your network and want to have a homogeneous network or, at least, a network with a reduced number of protocols, you can use MPTN connections to:

- Reduce the number of protocols.
- Save parallel homogenous protocols by piggy-backing on the SNA or IP backbone.
- Eliminate the need for multiple names for one workstation.
- Free the end users from concern about networking protocol is used in their environment.
- Extend the reach of applications across multiple networks by transport gateways.

MPTN Network Management

Multiprotocol networks present a major network management challenge. Multiprotocol networks introduce the problems involved in managing multiprotocol connections and datagrams. Errors are reported at the end-points to the application protocol, based on failures that occur at the transmission protocol. MPTN uses existing network management protocols.

Each environment is managed natively and separately by resource-specific managers. The transport user and transport provider are managed natively and separately using existing management products. In an MPTN environment, resources have as their native management protocol:

- SNA-Management Services (MS)

- Simple Network Management Protocol (SNMP)
- Common Management Information Protocol (CMIP)

Management of MPTN requires correlation among these different management protocols to identify problems that are signalled to a network management application as a result of incidents in the transport network.

Coordination management identifies problems that appear at the application as a result of problems in the transport layer. MPTN conforms to the SystemView structure with resource-specific managers, and coordination managers. SystemView coordination managers, such as the NetView program, consolidate information from SystemView resource-specific managers so the use of the various protocols can be consolidated.

SNA-MS Alerts are used by MPTN components to report errors. Transport protocol stacks are each managed natively by a resource-specific manager that understands the appropriate management protocol: SNA-MS, SNMP, or CMIP. These resource-specific managers report to a coordination manager, such as the NetView program, typically using the service point facilities. In some cases, there is translation of management protocols.

Sockets over SNA Network Management

The Sockets over SNA function can use existing NetView management capabilities. One difference between NetView management capabilities is the fact that socket applications are associated with a port that has a TCP/IP address. You can use the SXMAP utility provided with AnyNet/MVS to correlate the IP address to an SNA LU name. Once the SNA LU name is known, the network management of socket applications uses existing NetView capabilities.

Overview of the VTAM AnyNet Feature

The AnyNet family of products was introduced as the VTAM Multiprotocol Transport Feature (MPTF), in VTAM V3R4.2 for MVS/ESA.

In VTAM V4R2, the VTAM MPTF was renamed to the VTAM AnyNet Feature. It includes the following functions:

- SNA over TCP/IP
- Sockets over SNA
- Sockets over SNA Gateway.

For information about how to order the VTAM AnyNet Feature, refer to the product announcement letter for the VTAM V4R2 and the AnyNet Feature.

SNA over TCP/IP

This function, available in the MVS and OS/2 environments, enables SNA application programs to communicate over a TCP/IP network. In addition to supporting independent LU 6.2 communication, SNA over TCP/IP provides support for dependent logical unit communications, such as printers and emulators, if the host is defined as a dependent LU server and the VTAM-provided OS/2 dependent LU requester support is enabled at the workstation.

Sockets over SNA

This function, available in the MVS and OS/2 environments, enables application programs that use the C socket API to communicate over SNA networks with other application programs that also use the C socket interface. Sockets over SNA now supports LU 6.2 full-duplex conversations and VTAM's network-qualified name capability.

Sockets over SNA Gateway

This function, available in the OS/2 environment, connects TCP/IP and SNA networks to enable communication between socket applications. Socket applications on existing TCP/IP systems can communicate, without change, with socket applications on SNA systems that have AnyNet Sockets over SNA for MVS or AnyNet Sockets over SNA for OS/2 installed. In addition, AnyNet sockets gateways for OS/2 can be used to connect TCP/IP networks across an SNA network, thereby allowing communication between socket applications running on the separate TCP/IP networks.

Appendix L. Communications Manager/2's DLUR Support

The dependent LU server function provides dependent secondary logical unit (SLU) support by establishing an LU6.2 session pipe between an OS/2 dependent LU requester node (DLUR), and a dependent LU server node (DLUS). A DLUR is an APPN end node or an APPN network node that owns dependent LUs, but requests that a DLUS provide the SSCP services for those dependent LUs.

The VTAM-provided OS/2 DLUR for Communications Manager/2 provides full DLUR support, with the following exceptions:

- Downstream PUs
- SSCP takeover/giveback (ANS=CONT)
- DLUR/DLUS cross-subnetwork support
- SDDL
- XRF and XRF/Crypto.

In the future, IBM intends to provide the OS/2 DLUR function as part of Communications Manager/2.

This appendix describes:

- Downloading and unpacking the DLUR files
- Considerations for VTAM switched major node definitions

Note: Regardless of whether their connection to the DLUR is switched or leased, all DLUR-attached resources must be defined in a switched major node by using one of the following methods:

- Manually predefine them
 - Use the dynamic switched definition facility.
- Additions to Communications Manager/2 configuration commands
 - Sample NDF and RSP files.

Downloading the DLUR Files

Install the DLUR files by downloading the following files from the host to the root directory of the c: drive on your workstation.

1. **MVS** Download ISTIPDLR and ISTLDRM2 from the SYS1.SISTDAT1 library on the host. **VM** Download files ISTIPDLR AISTDAT1 and ISTLDRM2 AISTDAT1 from the 401 disk on the host. **VSE** Download ISTIPDLR.W and ISTLDRM2.W from the VTAM production library on the host.
2. **MVS** Download ISTIPRDM from the SYS1.SISTDAT2 library on the host. **VM** Download file ISTIPRDM AISTDAT1 from the 401 disk on the host. **VSE** Download ISTIPRDM.W from the VTAM production library on the host.

This is a README file containing DLUR installation and configuration information.

You can use any installed communication facility that enables you to transfer files to OS/2 from an MVS/ESA host or a VM/ESA host. Following are two examples of how you can download these files:

- Use the receive function of a Communications Manager/2 3270 emulation session. **MVS, VM** For example:

```
RECEIVE ISTIPDLR.RAM id:ISTIPDLR AISTDAT1 fm
RECEIVE LOADRAM2.EXE id:ISTLDRM2 AISTDAT1 fm
RECEIVE README.DLR id:ISTIPRDM AISTDAT1 fm (ASCII CRLF)
```

where id is the emulator session and fm is the filemode from which you are downloading.

VSE Or:

```
RECEIVE ISTIPDLR.RAM id:ISTIPDLR W (FILE=LIB L=vtamlib S=vtamsub BINARY
RECEIVE LOADRAM2.EXE id:ISTLDRM2 W (FILE=LIB L=vtamlib S=vtamsub BINARY
RECEIVE README.DLR id:ISTIPRDM W (FILE=LIB L=vtamlib S=vtamsub ASCII
```

where id is the CM/2 host emulator session for CICS and vtamlib and vtamsub identify your VTAM production sublibrary. For example, if PRD1.BASE is your VTAM production sublibrary, specify PRD1 for vtamlib and BASE for vtamsub.

For information on transferring files using 3270 sessions, refer to your *IBM Communications Manager/2 User's Guide* and *VSE/ESA Programming and Workstation Guide*.

- **MVS, VM** If TCP/IP is installed on the VM/ESA or MVS/ESA host and on your workstation, you can use FTP to download ISTIPDLR, ISTLDRM2, and ISTIPRDM. For information on setting up and using FTP on OS/2, refer to your *IBM TCP/IP for OS/2: User's Guide*.

Do the following when you download the DLUR files:

- Make sure you specify that files ISTIPDLR and ISTLDRM2 are in binary format before you run the download command. The README file, ISTIPRDM, is in EBCDIC format, not binary.
- Rename the following files after they are downloaded to OS/2:
 - Rename ISTIPDLR to ISTIPDLR.RAM.
 - Rename ISTLDRM2 to LOADRAM2.EXE.
 - Rename ISTIPRDM to README.DLR.

The following section refers to the DLUR files by their OS/2 names: ISTIPDLR.RAM and LOADRAM2.EXE. The README.DLR is DLUR information file and is not exploded.

Unpacking the DLUR Files into Communications Manager/2 Subdirectories

Make sure FFST and Communications Manager/2 and all associated processes are not active. It is best to reboot the machine or to not have Communications Manager/2 autostart before you begin this procedure.

The DLUR files should be on your c: drive as LOADRAM2.EXE and ISTIPDLR.RAM. The following procedure explains how to use LOADRAM2.EXE to unpack ISTIPDLR.RAM into individual DLUR files.

1. Back up and store both files. If any of the files are lost or damaged, you can restore them by running LOADRAM2 against ISTIPDLR again.
2. Make sure you are on the drive and in the subdirectory (if applicable) where LOADRAM2.EXE and ISTIPDLR.RAM reside.
3. Unpack ISTIPDLR.RAM into individual files using the following command:

Unpacking the DLUR Files into Communications Manager/2 Subdirectories

LOADRAM2 ISTIPDLR.RAM *target* /s [/d]

where:

LOADRAM2

name of the command that explodes ISTIPDLR.RAM into individual files.

ISTIPDLR.RAM

name of the file containing all packed files associated with DLUR Communications Manager/2.

target

drive where the individual DLUR files are stored when they are unpacked from ISTIPDLR.RAM. Specify the same drive on which Communications Manager/2 is installed, for example, c:.

/s instructs LOADRAM2 to restore files to the \CMLIB and \CMLIB\DLL subdirectories of the drive specified by *target*.

/d instructs LOADRAM2 to display files as they are unloaded. This parameter is optional.

4. The following file names are displayed when ISTIPDLR is exploded on the c: drive:

C:\CMLIB\DLL\DLUR.DLL

C:\CMLIB\DLR.MSG

C:\CMLIB\DLRH.MSG

If LOADRAM2 ran successfully, a return code of 0 should be displayed on your terminal. If a problem occurred while LOADRAM2 was exploding ISTIPDLR into individual files, one of the following error codes might be displayed on your terminal:

2	No files were found.
8	Insufficient memory.
15	An invalid drive was specified.
87	An invalid parameter was specified or incorrect syntax was used.
112	The target disk is full.
220	The present DOS version is not supported.
221	No files were found in the list file.
222	The macro file has the wrong format.
224	The macro file is too short.
225	Generally successful but no files were found to match at least one specification.

The following errors can also occur:

An error occurred while opening xxx

A file could not be transferred to the target directory. This error causes OS/2 to report *Path not found*. Ensure that the correct drive was specified.

Not transferred

A file could not be transferred to the target directory. This error can occur when reinstalling DLUR Communications Manager/2 if Com-

munications Manager/2 is active. Stop Communications Manager/2 and all associated processes, for example, subsystem management and message log formatters, and repeat the installation procedure.

Destination conflict

The target drive might not be specified correctly in the LOADRAM2 command.

5. Restart Communications Manager/2 and all associated processes.

Considerations for VTAM Switched Major Node Definitions for DLUR

This section describes the considerations for VTAM switched major node definition statements resulting from Communications Manager/2's implementation of DLUR.

- The IDBLK/IDNUM specified in the switched major node must match the NODE_ID value specified in the Communications Manager/2 RSP or NDF file, on the associated DEFINE_DEPENDENT_LU_SERVER statement.
- The IDBLK/IDNUM specified on the PU definition statement in the VTAM switched major node must match the value specified on the DLCADDR operand (when DLCADDR information is provided for IDBLK/IDNUM) on the associated PATH statement.
- The first DLCADDR entry defined for each PATH statement in the VTAM switched major node contains the DLC type information. For the Communications Manager/2's implementation of DLUR, the only DLC type supported is internal PU. This is coded as follows:

```
DLCADDR=(1,C,INTPU)
```

- The subsequent DLCADDR entries contain the DLC signaling information (that is, addresses and dial digits) for each DLC type. For the Communications Manager/2's implementation of DLUR, the DLC information will be either the IDBLK/IDNUM, CPNAME, or both. These entries are coded as follows:

```
DLCADDR=(2,X,idblk/idnum_value)
```

```
DLCADDR=(3,C,cpname_value)
```

- The LOCADDR coded on the LU definition statement in the VTAM switched major node must match the NAU_ADDRESS value, when defined for a specific DLUR-supported LU (defined in the Communications Manager/2 RSP file or the 3270 emulator panels).

Additions to Communications Manager/2 Configuration Commands for DLUR

This section describes the additions to the configuration commands resulting from Communications Manager/2's implementation of DLUR. The names shown here are the DEFAULT names defined in APPNV.CTL. You may customize any of these names to fit your requirements. You can change this section in APPNV.CTL, although this is not recommended. For more information on changing APPNV.CTL, see the *Communications Manager/2 Network Administration and Subsystem Management Guide*.

Communications Manager/2's implementation of DLUR requires the following changes to CMVERIFY.TXT. Note that there is no panel support for these changes.

- An addition to DEFINE_LOGICAL_LINK
- A new verb: DEFINE_DEPENDENT_LU_SERVER

Addition to DEFINE_LOGICAL_LINK

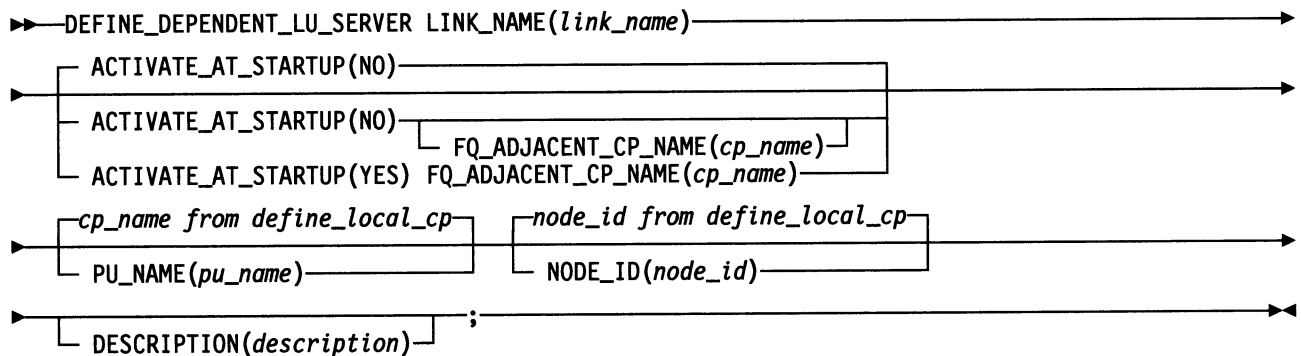
Add the following information to the explanation of the PU_NAME operand:

More than one link may be defined without a PU_NAME or with CP_NAME = PU_NAME, but only one of those links may have active SSCP-PU sessions at any moment in time. For PU_NAMES other than the CP, only one link can be defined with each PU_NAME.

More than one logical link or dependent LU server may be defined without a PU_NAME or with CP_NAME = PU_NAME, but only one such definition may have active SSCP-PU sessions at any moment in time. For PU_NAMES other than the CP, only one logical link or dependent LU server can be defined with each PU_NAME.

DEFINE_DEPENDENT_LU_SERVER

Full Syntax: The full syntax for DEFINE_DEPENDENT_LU_SERVER follows. The syntax for each operand is repeated in the “Operand Descriptions” section.



Operand Descriptions

The following section describes the operands you can code for DEFINE_DEPENDENT_LU_SERVER.

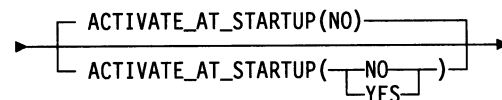
LINK_NAME

LINK_NAME(link_name)

specifies the name of the local logical link station. The *link_name* is a 1-8 byte type-a string.

If 3270 emulator sessions are to be used, the LINK_NAME must match the HOST_LINK_NAME (defined in the Communications Manager/2 RSP file or the 3270 emulator panels).

ACTIVATE_AT_STARTUP



specifies whether the connection to the DLUS should be established at Communications Manager/2 start time.

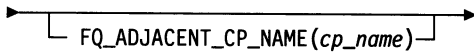
ACTIVATE_AT_STARTUP(NO)

The connection to the DLUS should not be established at Communications Manager/2 start time. NO is the default.

ACTIVATE_AT_STARTUP(YES)

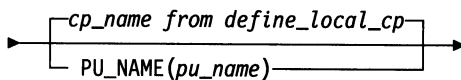
The connection to the DLUS should be established at Communications Manager/2 start time.

FQ_ADJACENT_CP_NAME



specifies the fully qualified DLUS control point name. The *cp_name* is a 3-17 byte type-a string. FQ_ADJACENT_CP_NAME is required if you code ACTIVATE_AT_STARTUP(YES). (That is, you want this station to establish the connection to the DLUS.) If you code ACTIVATE_AT_STARTUP(NO), you want DLUS to establish the connection, and FQ_ADJACENT_CP_NAME is optional.

PU_NAME



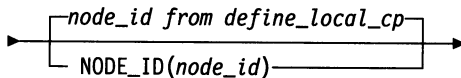
specifies the name of the local physical unit. The *pu_name* is a 1-8 byte type-a string.

PU_NAME is optional field. The default is the *cp_name* coded on the DEFINE_LOCAL_CP definition.

If the link is defined as the host focal point link (DEFINE_LOCAL_CP HOST_FP_LINK_NAME), the PU_NAME must be equivalent to the CP_NAME of this node.

More than one logical link or dependent LU server may be defined without a PU_NAME or with a CP_NAME the same as the PU_NAME, but only one such definition may have active SSCP-PU sessions at any given time. For PU_NAMES other than the CP, only one logical link or dependent LU server can be defined with each PU_NAME.

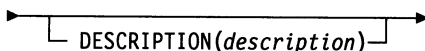
NODE_ID



specifies the IDBLK/IDNUM of the PU defined at the host. NODE_ID must be unique for each server. The *node_id* is a 5 or 8 digit hex string.

NODE_ID is optional field. The default is the *node_id* coded on the DEFINE_LOCAL_CP definition.

DESCRIPTION



is an optional 1-64 byte free format string describing this definition.

Sample DEFINE_DEPENDENT_LU_SERVER

```
DEFINE_DEPENDENT_LU_SERVER LINK_NAME(DLUS)
                           ACTIVATE_AT_STARTUP(NO)
                           FQ_ADJACENT_CP_NAME(APPN.DLUS)
                           PU_NAME(HOST2)
                           NODE_ID(X'05D00002')
                           DESCRIPTION(SECOND HOST VIA DLUS);
```

Sample Communications Manager/2 NDF File for DLUR

This section contains a sample Communications Manager/2 NDF file for the dependent LU requester illustrated in Figure 124 on page 503.

```
DEFINE_LOCAL_CP  FQ_CP_NAME(NETA.DLUR1      )
                  CP_ALIAS(BIGBLUE )
                  NAU_ADDRESS(INDEPENDENT_LU)
                  NODE_TYPE(EN)
                  NODE_ID(X'05D00002')
                  NW_FP_SUPPORT(NONE)
                  HOST_FP_SUPPORT(NO)
                  MAX_COMP_LEVEL(LZ9)
                  MAX_COMP_TOKENS(44);

DEFINE_LOGICAL_LINK LINK_NAME(LINK0001)
                    ADJACENT_NODE_TYPE(NN)
                    PREFERRED_NN_SERVER(NO)
                    DLC_NAME(IBMTRNET)
                    ADAPTER_NUMBER(0)
                    DESTINATION_ADDRESS(X'40000777711604')
                    ETHERNET_FORMAT(NO)
                    CP_CP_SESSION_SUPPORT(YES)
                    SOLICIT_SSCP_SESSION(NO)
                    ACTIVATE_AT_STARTUP(YES)
                    USE_PUNAME_AS_CPNAME(NO)
                    LIMITED_RESOURCE(NO)
                    LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                    MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                    EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                    COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                    COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                    SECURITY(USE_ADAPTER_DEFINITION)
                    PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_DEPENDENT_LU_SERVER LINK_NAME(VLINK1 )
                           FQ_ADJACENT_CP_NAME(NETA.VTAMB      )
                           PU_NAME(PU4      )
                           NODE_ID(X'05D00001')
                           ACTIVATE_AT_STARTUP(NO);

DEFINE_DEPENDENT_LU_SERVER LINK_NAME(VLINK2 )
                           FQ_ADJACENT_CP_NAME(NETA.VTAMB      )
                           PU_NAME(PU5      )
```

Sample Communications Manager/2 RSP File for DLUR

```

                                NODE_ID(X'05D00002')
                                ACTIVATE_AT_STARTUP(NO);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(NETA.APPCAP06    )
                   DESCRIPTION(King Tut)
                   PARTNER_LU_ALIAS(KINGTUT)
                   PARTNER_LU_UNINTERPRETED_NAME(KINGTUT )
                   MAX_MC_LL_SEND_SIZE(4096)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(NO);

DEFINE_MODE  MODE_NAME(BATCH    )
             DESCRIPTION(single session mode)
             COS_NAME(#CONNECT)
             DEFAULT_RU_SIZE(YES)
             MAX_RU_SIZE_UPPER_BOUND(4096)
             RECEIVE_PACING_WINDOW(7)
             MAX_NEGOTIABLE_SESSION_LIMIT(1)
             PLU_MODE_SESSION_LIMIT(1)
             MIN_CONWINNERS_SOURCE(0)
             COMPRESSION_NEED(PROHIBITED)
             PLU_SLU_COMPRESSION(NONE)
             SLU_PLU_COMPRESSION(NONE);

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                 DEFAULT_MODE_NAME(BLANK)
                 MAX_MC_LL_SEND_SIZE(32767)
                 DIRECTORY_FOR_INBOUND_ATTACHES(*)
                 DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                 DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                 DEFAULT_TP_CONV_SECURITY_RQD(NO)
                 MAX_HELD_ALERTS(10);

START_ATTACH_MANAGER;
```

Sample Communications Manager/2 RSP File for DLUR

This appendix contains a sample Communications Manager/2 RSP file for the dependent LU requester illustrated in Figure 124 on page 503.

```
*****
* Keywords and values are documented in:
* Response File Reference (Online book) found on the Response File Samples
* Diskette.
*
* To use this response file to install or configure Communications Manager/2
* using the CMSETUP /R or CMLAN /R command, values for one or more of these
* keywords must be provided:
  CMUPDATETYPE = 4
  CMUSERCFG = d:\cmlib\SAMPCMGR.cfg
* CMMODELCFG =
*****

* Installation keywords:

  CMTARGET = D:
  CMINSTALLFOLDERS = 1
```

```

CMINSTALLSSM = 1
CMINSTALLKEYBDREMAP = 1
CMINSTALLROPS = 0
CMINSTALLPD = 1
CMINSTALLCOMMANDREFERENCE = 1
CMINSTALLMESSAGEREFERENCE = 1
CMINSTALLKEYLOCK = 1
CMINSTALLCONFIGDIALOG = 1
CMINSTALLOVERVIEW = 1
CMINSTALLGLOSSARY = 0
CMINSTALLPDGUIDE = 0
CMSERVER = 1
CMINSTALLUPM = 0
CONFIGSYSTEMMAXDUMPS = 32
CONFIGSYSTEMDUMPPATH = C:\OS2\SYSTEM\
CONFIGAPPLMAXDUMPS = 32
CONFIGAPPLDUMPPATH = C:\OS2\SYSTEM\
CONFIGMSGLOGNAME = C:\OS2\SYSTEM\OS2MLOG.DAT
CONFIGWSID = PRGWKSTN
CONFIGWSSERIAL1 = 00
CONFIGWSSERIAL2 = 0000000
CONFIGWSTYPE1 = 8565
CONFIGWSTYPE2 = 000
CONFIGDISPLAYMSG = 1
CMWORKSTATIONTYPE = 1

```

* Configuration keywords:

```

3270_COLOR = (
  NAME = 3270COLR
  B_NORMAL_UNPROT = 1
  B_INTENSE_UNPROT = 1
  B_NORMAL_PROT = 1
  B_INTENSE_PROT = 1
  B_BLUE = 1
  B_GREEN = 1
  B_PINK = 1
  B_RED = 1
  B_TURQUOISE = 1
  B_WHITE = 1
  B_YELLOW = 1
  B_DEFAULT_HILITE = 1
  B_DEFAULT_NO_HILITE = 1
  B_BLINK = 8
  B_OIA = 3
  F_NORMAL_UNPROT = 3
  F_INTENSE_UNPROT = 5
  F_NORMAL_PROT = 4
  F_INTENSE_PROT = 8
  F_BLUE = 4
  F_GREEN = 3
  F_PINK = 14
  F_RED = 5
  F_TURQUOISE = 12
  F_WHITE = 8
  F_YELLOW = 15
  F_DEFAULT_HILITE = 8
  F_DEFAULT_NO_HILITE = 3

```

```

        F_BLINK = 1
        F_OIA = 1
    )

* 3270_CONNECT is not configured.

3270_SESSION = (
    NAME = A
    SESSION_TYPE = 1
* COMMENT is either not used or not configured.
    LONG_SESSION_NAME = DLR1C
    SESSION_AUTOSTART = 0
    ACTIVATE_PS_PRINT = 0
    SHOW_WINDOW = 1
    SPACE_SIZE = 1
    AT_KEYBOARD_NAME = ACSCATUS
    ENHANCED_KEYBOARD_NAME = ACSCENUS
    COLOR_NAME = 3270COLR
    PRINT_BUFFER_SIZE = 1920
    CHAR_MAP_TYPE = 3
    SUBSTITUTION_CHAR = NULL
    DATA_CONFID = 1
    SELECTIVE_PC_DATA_CONFID = 0
    DATA_CONFID_ID = BLANK
    XFER_SIZE = 0
    PS_SIZE_ROW = 25
    PS_SIZE_COLUMN = 80
    UNSUP_CONTROL_CODE = 2
    EMU_TYPE = 1
    NAU_ADDRESS = 1
* LOGICAL_TERM_NUMBER is either not used or not configured.
    DATA_COMPRESSION = 1
    HOST_LINK_NAME = VLINK1
    HOST_ALARM = 2
    PROTECTED_FIELD_ALARM = 2
    SCREEN_UPDATE_ALARM = 2
    TITLE_BAR = 2
    DISPLAY_ROW_COLUMN = 1
    DISPLAY_ICON = 2
    MIN_MAX_STATE = 1
    AUTO_FONT = 2
    MENU_BAR = 1
    HOTSPOT_CMDNN = 1
    HOTSPOT_FNN = 1
    HOTSPOT_FPNN = 1
    HOTSPOT_NN = 1
    HOTSPOT_PFNN = 1
    HOTSPOT_SELECT_STRING = 1
* CCD0420: No PRINTER fields are configured. SESSION_TYPE must equal 2, or
* ACTIVATE_PS_PRINT must equal 1, for PRINTER fields to be configured.
)

3270_SESSION = (
    NAME = B
    SESSION_TYPE = 1
* COMMENT is either not used or not configured.
    LONG_SESSION_NAME = DLR2C

```

```

SESSION_AUTOSTART = 0
ACTIVATE_PS_PRINT = 0
SHOW_WINDOW = 1
SPACE_SIZE = 1
AT_KEYBOARD_NAME = ACSCATUS
ENHANCED_KEYBOARD_NAME = ACSCENUS
COLOR_NAME = 3270COLR
PRINT_BUFFER_SIZE = 1920
CHAR_MAP_TYPE = 3
SUBSTITUTION_CHAR = NULL
DATA_CONFID = 1
SELECTIVE_PC_DATA_CONFID = 0
DATA_CONFID_ID = BLANK
XFER_SIZE = 0
PS_SIZE_ROW = 25
PS_SIZE_COLUMN = 80
UNSUP_CONTROL_CODE = 1
EMU_TYPE = 1
NAU_ADDRESS = 1
* LOGICAL_TERM_NUMBER is either not used or not configured.
DATA_COMPRESSION = 1
HOST_LINK_NAME = VLINK2
HOST_ALARM = 1
PROTECTED_FIELD_ALARM = 1
SCREEN_UPDATE_ALARM = 1
TITLE_BAR = 1
DISPLAY_ROW_COLUMN = 2
DISPLAY_ICON = 2
MIN_MAX_STATE = 2
AUTO_FONT = 1
MENU_BAR = 1
HOTSPOT_CMDNN = 1
HOTSPOT_FNN = 1
HOTSPOT_FPNN = 1
HOTSPOT_NN = 1
HOTSPOT_PFNN = 1
HOTSPOT_SELECT_STRING = 1
* CCD0420: No PRINTER fields are configured. SESSION_TYPE must equal 2, or
* ACTIVATE_PS_PRINT must equal 1, for PRINTER fields to be configured.
)

* 5250_COLOR is not configured.

* 5250_SESSION is not configured.

* ACDI_PORT is not configured.

* ACDI_REDIRECT is not configured.

* ACDI_SPECIAL_CHARACTERS is not configured.

AT_KEYBOARD = (
    NAME = ACSCATUS
)

```

```
ENHANCED_KEYBOARD = (
    NAME = ACSCENUS
)
```

```
ENHANCED_KEYBOARD = (
    NAME = ACSOENUS
)
```

* FT_CICS is not configured.

* FT_MISC is not configured.

* FT_TSO is not configured.

* FT_VM is not configured.

* IDLC_DLC is not configured.

```
LAN_DLC = (
    NAME = 0
    ADAPTER_TYPE = 2
    MAX_LINK_STATION = 010
    PER_INCOMING_CALLS = 010
    FREE_LINK = 0
    MAX_I_FIELD_SIZE = 1929
    SEND_COUNT = 4
    RECEIVE_COUNT = 4
    CASM_LAN_ID = USIBMNR
    SEND_ALERT = 0
    LINK_ESTABLISHMENT_RETRANSMISSION = 30
    MAX_ACTIVATION_ATTEMPTS = 32
    LOCAL_SAP = 04
    RETRANSMISSION_THRESHOLD = 32
)
```

* SDLC_DLC is not configured.

* SRPI_SERVER is not configured.

* SRPI_SVRBASE is not configured.

* TWINAX_DLC is not configured.

```
WORKSTATION = (
    COMMENT = BASIC CONFIGURATION FILE
    * TRANS_TABLE is either not used or not configured.
    LOAD_SNA_APPC = 1
    LOAD_SRPI = 1
    LOAD_X25_API = 1
    LOAD_ACDI = 1
)
```



```

    USE_ACDI_API = 0
    USE_ARTIC_ADAPTER_WITHOUT_X25 = 0
    USE_LAN_PROTOCOLS_USING_ISDN = 0
)

* X25_DLC is not configured.

* X25_LDIR is not configured.

* X25_LINK is not configured.

* X25_LINK_PVC is not configured.

* X25_RDIR_NSNA_PVC is not configured.

* X25_RDIR_NSNA_SVC is not configured.

* X25_RDIR_SNA_PVC is not configured.

* X25_RDIR_SNA_SVC is not configured.

* X25_ROUTE is not configured.

* ACTIVATE_LOGICAL_LINKS is not configured.

ATTACH_MANAGER = (
    START = 1
)

* CNOS_FQ_PLU_NAME is not configured.

* CNOS_PLU_ALIAS is not configured.

* CONNECTION_NETWORK is not configured.

* COS is not configured.

* CPIC_SIDE_INFO is not configured.

* DEFAULT_SOC is not configured.

```

* EXPLICIT_SOC is not configured.

* GATEWAY_HOST is not configured.

* GATEWAY_WORKSTATION is not configured.

```
LOCAL_CP = (
    NAME = NETA.DLUR1
    * COMMENT is either not used or not configured.
    CP_ALIAS = BIGBLUE
    * HOST_FP_LINK_NAME is either not used or not configured.
    HOST_FP_SUPPORT = 0
    NAU_ADDRESS = 0
    NODE_ID = 05D00002
    NODE_TYPE = 3
    NW_FP_SUPPORT = 0
    * NW_SERVER_NAME is either not used or not configured.
    MAX_COMPRESSION_LEVEL = 2
    MAX_COMPRESSION_TOKENS = 44
)
```

* LOCAL_LU is not configured.

```
LOGICAL_LINK = (
    NAME = LINK0001
    * COMMENT is either not used or not configured.
    DLC_NAME = IBMTRNET
    ACTIVATE_AT_STARTUP = 1
    CP_CP_SESSION_SUPPORT = 1
    ADAPTER_NUMBER = 0
    ADJACENT_NODE_TYPE = 2
    PREFERRED_NN_SERVER = 0
    DESTINATION_ADDRESS = 40000777711604
    * INIT_WITH_SNRN is either not used or not configured.
    ETHERNET_FORMAT = 0
    * OUTGOING_DIR_ENTRY_NAME is either not used or not configured.
    * X25_DIR_ENTRY_NAME is either not used or not configured.
    * FQ_ADJACENT_CP_NAME is either not used or not configured.
    EFFECTIVE_CAPACITY = -1
    COST_PER_BYTE = -1
    COST_PER_CONNECT_TIME = -1
    PROPAGATION_DELAY = -1
    SECURITY = -1
    USER_DEFINED_1 = -1
    USER_DEFINED_2 = -1
    USER_DEFINED_3 = -1
    LIMITED_RESOURCE = 0
    LINK_STATION_ROLE = -1
    * PU_NAME is either not used or not configured.
    SOLICIT_SSCP_SESSION = 0
    * NODE_ID is either not used or not configured.
    * ADJACENT_NODE_ID is either not used or not configured.
    MAX_ACTIVATION_ATTEMPTS = 0
    USE_PUNAME_AS_CPNAME = 0
    * IDLC_T200 is either not used or not configured.
    * IDLC_N200 is either not used or not configured.
```

```

*   IDLC_N201 is either not used or not configured.
*   IDLC_T203 is either not used or not configured.
*   IDLC_K is either not used or not configured.
*   IDLC_NW is either not used or not configured.
)

LOGICAL_LINK = (
    NAME = VLINK1
*   COMMENT is either not used or not configured.
    DLC_NAME = DLUSVR
    ACTIVATE_AT_STARTUP = 0
    CP_CP_SESSION_SUPPORT = 0
    ADAPTER_NUMBER = 0
    ADJACENT_NODE_TYPE = 0
*   PREFERRED_NN_SERVER is either not used or not configured.
*   DESTINATION_ADDRESS is either not used or not configured.
*   INIT_WITH_SNRM is either not used or not configured.
    ETHERNET_FORMAT = 0
*   OUTGOING_DIR_ENTRY_NAME is either not used or not configured.
*   X25_DIR_ENTRY_NAME is either not used or not configured.
    FQ_ADJACENT_CP_NAME = NETA.VTAMB
    EFFECTIVE_CAPACITY = -1
    COST_PER_BYTE = -1
    COST_PER_CONNECT_TIME = -1
    PROPAGATION_DELAY = -1
    SECURITY = -1
    USER_DEFINED_1 = -1
    USER_DEFINED_2 = -1
    USER_DEFINED_3 = -1
    LIMITED_RESOURCE = -1
    LINK_STATION_ROLE = -1
    PU_NAME = PU4
    SOLICIT_SSCP_SESSION = 1
    NODE_ID = 05D00001
*   ADJACENT_NODE_ID is either not used or not configured.
    MAX_ACTIVATION_ATTEMPTS = 0
    USE_PUNAME_AS_CPNAME = 0
*   IDLC_T200 is either not used or not configured.
*   IDLC_N200 is either not used or not configured.
*   IDLC_N201 is either not used or not configured.
*   IDLC_T203 is either not used or not configured.
*   IDLC_K is either not used or not configured.
*   IDLC_NW is either not used or not configured.
)

LOGICAL_LINK = (
    NAME = VLINK2
*   COMMENT is either not used or not configured.
    DLC_NAME = DLUSVR
    ACTIVATE_AT_STARTUP = 0
    CP_CP_SESSION_SUPPORT = 0
    ADAPTER_NUMBER = 0
    ADJACENT_NODE_TYPE = 0
*   PREFERRED_NN_SERVER is either not used or not configured.
*   DESTINATION_ADDRESS is either not used or not configured.
*   INIT_WITH_SNRM is either not used or not configured.

```

```

    ETHERNET_FORMAT = 0
*   OUTGOING_DIR_ENTRY_NAME is either not used or not configured.
*   X25_DIR_ENTRY_NAME is either not used or not configured.
    FQ_ADJACENT_CP_NAME = NETA.VTAMB
    EFFECTIVE_CAPACITY = -1
    COST_PER_BYTE = -1
    COST_PER_CONNECT_TIME = -1
    PROPAGATION_DELAY = -1
    SECURITY = -1
    USER_DEFINED_1 = -1
    USER_DEFINED_2 = -1
    USER_DEFINED_3 = -1
    LIMITED_RESOURCE = -1
    LINK_STATION_ROLE = -1
    PU_NAME = PU5
    SOLICIT_SSCP_SESSION = 1
    NODE_ID = 05D00002
*   ADJACENT_NODE_ID is either not used or not configured.
    MAX_ACTIVATION_ATTEMPTS = 0
    USE_PUNAME_AS_CPNAME = 0
*   IDLC_T200 is either not used or not configured.
*   IDLC_N200 is either not used or not configured.
*   IDLC_N201 is either not used or not configured.
*   IDLC_T203 is either not used or not configured.
*   IDLC_K is either not used or not configured.
*   IDLC_NW is either not used or not configured.
)

LUA = (
    NAME = LUA1
*   COMMENT is either not used or not configured.
    HOST_LINK_NAME = VLINK1
    NAU_ADDRESS = 25
)

* LU_LU_PASSWORD is not configured.

MODE = (
    NAME = BATCH
    COMMENT = single session mode
    MAX_RU_SIZE_UPPER_BOUND = 4096
    COS_NAME = #CONNECT
    DEFAULT_RU_SIZE = 1
    MAX_NEGOTIABLE_SESSION_LIMIT = 1
    MIN_CONWINNERS_SOURCE = 0
    PLU_MODE_SESSION_LIMIT = 1
    RECEIVE_PACING_WINDOW = 7
    COMPRESSION_NEED = 0
    PLU_SLU_COMPRESSION = 0
    SLU_PLU_COMPRESSION = 0
)

PARTNER_LU = (
    NAME = NETA.APPCAP06
    COMMENT = King Tut
    PARTNER_LU_ALIAS = KINGTUT
    PARTNER_LU_UNINTERPRETED_NAME = KINGTUT

```

```
    CONV_SECURITY_VERIFICATION = 0
    MAX_MC_LL_SEND_SIZE = 4096
    PARALLEL_SESSION_SUPPORT = 0
)

* PARTNER_LU_LOCATION is not configured.

* REMOTE_FOCAL_POINT is not configured.

SNA_DEFAULTS = (
*   COMMENT is either not used or not configured.
    DEFAULT_TP_CONV_SECURITY_RQD = 0
    DEFAULT_TP_OPERATION = 2
    DEFAULT_TP_PROGRAM_TYPE = 0
    DIRECTORY_FOR_INBOUND_ATTACHES = *
    IMPLICIT_INBOUND_PLU_SUPPORT = 1
    DEFAULT_MODE_NAME = BLANK
*   DEFAULT_LOCAL_LU_ALIAS is either not used or not configured.
    MAX_HELD_ALERTS = 10
    MAX_MC_LL_SEND_SIZE = 32767
)

* TP is not configured.

* USER_ID_PASSWORD is not configured.

* INCOMING_CALL_DIR is not configured.

* OUTGOING_CALL_DIR is not configured.

* SWITCH_TYPE is not configured.
```

Glossary, Bibliography, and Index

Glossary	725
Bibliography	727
VTAM V4R2 Publications	727
Softcopy Information	727
Marketing Information	727
Planning	727
Installation, Resource Definition, and Tuning	729
Operation	729
Customization	730
Writing Application Programs	731
Diagnosis	731
VTAM AnyNet Feature for V4R2 for MVS/ESA	732
MPTN Architecture Publications	732
APPC Application Suite Feature for V4R2 for MVS/ESA	732
Related Publications	732
Index	733

Glossary

For definitions of the terms and abbreviations used in this book, refer to the *VTAM Glossary*.

Bibliography

VTAM V4R2 Publications

Following are descriptions of the books in the VTAM V4R2 library for the MVS/ESA, VM/ESA, and VSE/ESA operating systems. The books are arranged here according to the tasks they describe:

- Planning
- Installation, Resource Definition, and Tuning
- Customization
- Operation
- Writing Application Programs
- Diagnosis
- Quick Reference.

The complete set of unlicensed books in this section can be ordered using a single order number, SBOF-4394 for MVS, SBOF-7001 for VM, and SBOF-7002 for VSE.

Softcopy Information

IBM Networking Softcopy Collection Kit CD-ROM (SK2T-6012):

The softcopy library contains softcopy versions of the licensed and unlicensed books for VTAM V4R2 and the VTAM AnyNet Feature for V4R2 for MVS/ESA.

All of the unlicensed and licensed VTAM books described in this section are available in softcopy on this CD-ROM. These softcopy files can be read using any of the IBM BookManager READ programs. They can also be read with the IBM Library Reader program shipped on this CD.

In addition, this CD contains the Online Message Facility. The Online Message Facility is an OS/2 program that provides online access to information from *VTAM Messages and Codes* and other BookManager softcopy books. The facility helps network operators and system programmers operate and diagnose problems without interrupting those tasks.

The CD also contains softcopy of the unlicensed books of many other products.

Marketing Information

A Networking Overview and the following IBM Networking Previews are available:

- AnyNet
- VTAM.

Ask your IBM marketing representative for more information.

Planning

Planning for NetView, NCP, and VTAM

(SC31-7122): This book helps you plan for new products or for migrating to new releases of networking products. It describes product functions, explains benefits you can gain from using them in different situations, and address cross-product implications. The book contains cross-task reference information and storage estimates.

Planning for Integrated Networks

(SC31-7123): This book helps you plan for SNA (subarea and APPN) and TCP/IP networks. It includes discussion of protocol strategies, migration scenarios, processing goals, and management considerations.

Planning Aids: Pre-Installation Planning Checklist for NetView, NCP, and VTAM

(SX75-0092): This checklist identifies important tasks to consider and complete before you begin to install these product. The document can be reproduced and folded to fit easily in a pocket or folder for quick reference and easy portability.

VTAM Licensed Program Specifications

(GC31-6490): This flyer is the warranty for VTAM and includes:

- A list of new functions
- Descriptions of VTAM features
- Machine requirements
- Programming requirements.

VTAM Release Guide for MVS/ESA

(GC31-6492): This guide provides an overview of the new functions in VTAM V4R2 and includes:

- Advantages of new functions
- Planning considerations for new functions
- Effect of new functions on existing functions
- Changes to commands, definition statements, and messages
- Programming requirements, such as the release of NCP required.

VTAM Release Guide for VM/ESA

(GC31-8089): This guide provides an overview of the new functions in VTAM V4R2 and includes:

- Advantages of new functions
- Planning considerations for new functions
- Effect of new functions on existing functions
- Changes to commands, definition statements, and messages
- Programming requirements, such as the release of NCP required.

VTAM Release Guide for VSE/ESA

(GC31-8090): This guide provides an overview of the new functions in VTAM V4R2 and includes:

- Advantages of new functions
- Planning considerations for new functions
- Effect of new functions on existing functions
- Changes to commands, definition statements, and messages
- Programming requirements, such as the release of NCP required.

VTAM Migration Guide for MVS/ESA

(GC31-6491): This guide helps you upgrade VTAM V4R1, V3R4.2, V3R4.1, V3R4, or V3R3 to VTAM V4R2. It includes:

- Planning to upgrade to VTAM V4R2
 - Upward and downward compatibility
 - Software and hardware requirements
 - Storage requirements
 - Impacts of new functions and enhancements performed without changes to user interfaces
 - Changes to installation process
- Upgrading user interfaces to VTAM V4R2
 - Changes to start options
 - Changes to buffer pools
 - Changes to definition statements
 - Changes to IBM-supplied default user-definable tables and modules
 - Changes to user-definable table macroinstructions
 - Changes to commands
 - Changes to messages

- Changes to VTAM application programming interface
- Changes to installation-wide exit routines
- Changes to control blocks
- Implementing optional functions and enhancements introduced in VTAM V4R2
 - Overview of each new function and enhancement introduced since VTAM V3R3
 - Pointers to other books in the library where implementation details can be found.

VTAM Migration Guide for VM/ESA

(GC31-8071): This guide helps you upgrade VTAM V3R4.1 or V3R4 to VTAM V4R2. It includes:

- Planning to upgrade to VTAM V4R2
 - Upward and downward compatibility
 - Software and hardware requirements
 - Storage requirements
 - Impacts of new functions and enhancements performed without changes to user interfaces
 - Changes to installation process
- Upgrading user interfaces to VTAM V4R2
 - Changes to start options
 - Changes to buffer pools
 - Changes to definition statements
 - Changes to IBM-supplied default user-definable tables and modules
 - Changes to user-definable table macroinstructions
 - Changes to commands
 - Changes to messages
 - Changes to VTAM application programming interface
 - Changes to installation-wide exit routines
 - Changes to control blocks
- Implementing optional functions and enhancements introduced in VTAM V4R2
 - Overview of each new function and enhancement introduced since VTAM V3R4
 - Pointers to other books in the library where implementation details can be found.

VTAM Migration Guide for VSE/ESA

(GC31-8072): This guide helps you upgrade VTAM V3R4, V3R3, or V3R2 to VTAM V4R2. It includes:

- Planning to upgrade to VTAM V4R2
 - Upward and downward compatibility
 - Software and hardware requirements
 - Storage requirements
 - Impacts of new functions and enhancements performed without changes to user interfaces
 - Changes to installation process
- Upgrading user interfaces to VTAM V4R2
 - Changes to start options
 - Changes to buffer pools
 - Changes to definition statements

- Changes to IBM-supplied default user-definable tables and modules
- Changes to user-definable table macroinstructions
- Changes to commands
- Changes to messages
- Changes to VTAM application programming interface
- Changes to installation-wide exit routines
- Changes to control blocks
- Implementing optional functions and enhancements introduced in VTAM V4R2
 - Overview of each new function and enhancement introduced since VTAM V3R2
 - Pointers to other books in the library where implementation details can be found.

Estimating Storage for VTAM

(SK2T-2007): This interactive program helps you estimate the storage requirements for VTAM. The diskette includes:

- Step-by-step procedures
- Formulas used to calculate storage.

VTAM Overview for VM/ESA and

VSE/ESA (GC31-8114): This document is designed to be used with each book of the VTAM library. It helps you understand which functions are included with each package available for VTAM V4R2 for VM/ESA and VSE/ESA.

- VTAM V4R2 Client/Server
- VTAM V4R2 MultiDomain
- VTAM V4R2 InterEnterprise.

It also provides instructions for how to order a particular package.

VTAM Glossary (GC31-6558): This glossary defines terms and abbreviations for VTAM and related products. It includes information from the IBM *Dictionary of Computing*, SC20-1699.

Installation, Resource Definition, and Tuning

VTAM Network Implementation Guide

(SC31-6494): This book presents the major concepts involved in implementing a VTAM network, and includes:

- Buffer pools, slowdown, pacing, storage considerations
- Implementation considerations

- Installation procedures
- Sample major node definitions
- Migration considerations
- Tables and filters
- TSO, VSCS, VCNS, and other programs that run with VTAM
- Tuning procedures
- VTAM start options.

Use this book in conjunction with the *VTAM Resource Definition Reference*.

VTAM Resource Definition Reference

(SC31-6498): This book describes each VTAM definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect VTAM. The information includes:

- IBM-supplied default tables (logon mode, USS and X.25)
- DTIGEN macroinstruction (VSCS start options)
- Major node definitions
- User-defined tables and filters
- VTAM start options.

If you are unfamiliar with the major concepts involved in implementing a VTAM network, use this book in conjunction with the *VTAM Network Implementation Guide*.

VTAM Resource Definition Samples

(SC31-6499): This book contains sample definitions to help you implement VTAM functions in your networks, and includes sample major node definitions.

Use this book in conjunction with the *VTAM Network Implementation Guide* and *VTAM Resource Definition Reference*.

Operation

VTAM Operation (SC31-6495): This book serves as a reference for programmers and operators requiring detailed information about specific operator commands. The information includes:

- VTAM commands and start options
- Logon manager commands
- DISPLAY output examples (messages received)
- VSCS commands.

VTAM Operation Quick Reference

(SX75-0205): This book contains essential information about VTAM and VSCS operator commands.

VTAM Messages and Codes

(SC31-6493): This book describes messages, codes, and other information associated with VTAM messages and includes:

- Messages:
 - ELM messages for logon manager
 - IKT messages for TSO/VTAM
 - IST messages for VTAM network operators
 - ISU messages for sockets-over-SNA
 - USS messages
 - VSCS messages
- Codes and other information that display in VTAM messages:
 - Abend codes
 - Command and RU types in VTAM messages
 - Node and ID types in VTAM messages
 - Return codes for macroinstructions including ACB OPEN and CLOSE macroinstruction error fields, RTNCD-FDB2 return code combinations, and LU 6.2 RCPRI-RCSEC return codes
 - Sense codes including VTAM sense code hints, SNA sense field values for RPL-based macroinstructions, and 3270 SNA and non-SNA device sense fields
 - Status codes including resource status and session state codes
 - Wait state event codes and IDs
- Supplemental message-related information:
 - Message additions, deletions, and changes
 - Message flooding prevention
 - Message groups and subgroups
 - Message routing and suppression including descriptor codes, routing codes, and suppression levels for ELM, IKT, IST, and ISU messages
 - Message text and description formats
 - Message text of MSGLVL option messages including general information on the MSGLVL option
 - Message text of all VTAM network operator messages including variable field lengths
 - Online Message Facility.

Using IBM CommandTree/2

(SC31-7013): IBM CommandTree/2 is a workstation product that enables an operator to construct commands and send them to a specified destination for processing. The VTAM command set library includes:

- VTAM commands
- Logon manager commands
- Help for commands and start options.
- VSCS commands

Customization

VTAM Customization (LY43-0063): This book enables you to customize VTAM, and includes:

- Communication network management (CNM) routing table
- Installing tables and modules in VM
- Logon-interpret routine requirements
- Logon manager installation-wide exit routine for the CLU search exit
- VSCS data manipulation installation-wide exit routine
- TSO/VTAM installation-wide exit routines
- VTAM installation-wide exit routines:
 - Command verification exit (ISTCMMND)
 - Configuration services XID exit (ISTEXCCS) with description of IBM-supplied default exit
 - Directory services management exit (ISTEXCDM)
 - Generic resource resolution exit (ISTEXCGR)
 - SDDL exit (ISTEXCSD) with description of IBM-supplied default exit
 - Session accounting exit (ISTAUCAG)
 - Session authorization exit (ISTAUCAT)
 - Session management exit (ISTEXCAA) with example
 - TPRINT processing exit (ISTRAEUE)
 - USERVAR exit (ISTEXCUV) with description of IBM-supplied default exit
 - Virtual route pacing window size calculation exit (ISTPUCWC)
 - Virtual route selection exit (ISTEXCVR).

Writing Application Programs

VTAM Programming (SC31-6496): This book describes how to use VTAM macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain. The information includes:

- API concepts
 - Cryptography
 - RUs and exchanges
 - Session establishment and termination
- BIND area format
- Communication Network Management Interface
- Dictionary of VTAM macroinstructions
- OPEN or CLOSE errors
- Operating system differences
- Program Operator Coding requirements
- RAPI DSECTs and control block mappings
- RAPI global variables
- Resource-identification and access-method-support vector lists
- RPL-based macroinstructions
- RPL RTNCD,FDB2 codes
- User exit routines.

VTAM Programming for LU 6.2

(SC31-6497): This book describes the VTAM LU 6.2 programming interface for host application programs. This book applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this book.) The information includes:

- Allocating and deallocating conversations
- APPCCMD macroinstructions and LU 6.2 DSECTs
- BIND image and response and ISTDBIND
- Conversation states
- Description and use of the following control blocks:
 - CNOS session limits control block
 - DEFINE control block
 - DISPLAY control block
 - RESTORE control block
- FMH-5 and PIP data
- LU 6.2 global variables
- Resource-identification and access-method-support vector lists
- RCPRI,RCSEC codes
- Sample program for retrieving RESTORE information
- Sample VTAM LU 6.2 application program
- Session- and conversation-level security and data encryption

- Sending and receiving data
- Sense codes for FMH-7 and UNBIND
- Summary of register usage
- Sync point services
- User exit routines.

Diagnosis

VTAM Diagnosis (LY43-0065): This book helps you identify a VTAM problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation. The information includes:

- Command syntax for running traces and collecting and analyzing dumps
- VIT entries
- Procedures for collecting documentation (VTAM, VSCS, TSO)
- VTAM internal trace and VIT analysis tool
- FFST Probes
- Channel programs
- Flow diagrams
- Procedures for locating buffer pools
- VSCS dump and traces
- CPCB operation codes
- Storage and control block ID codes
- PIU discard reason codes
- Offset names and locations for VTAM buffer pools

VTAM Data Areas for MVS/ESA

(LY43-0064): This book describes VTAM data areas and can be used to read a VTAM dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

VTAM Data Areas for VM/ESA

(LY43-0103): This book describes VTAM data areas and can be used to read a VTAM dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

VTAM Data Areas for VSE/ESA

(LY43-0104): This book describes VTAM data areas and can be used to read a VTAM dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

VTAM AnyNet Feature for V4R2 for MVS/ESA

VTAM AnyNet Feature for V4R2: Guide to Sockets over SNA Gateway for OS/2 (SC31-6528): This guide provides information to help you install, configure, use, and diagnose the sockets-over-SNA-gateway function for OS/2. This function allows socket applications running on a TCP/IP network to communicate with socket applications running on an SNA network.

VTAM AnyNet Feature for V4R2: Guide to Sockets over SNA (SC31-6526): This guide provides information to help you install, configure, use, and diagnose Sockets over SNA. It also provides information to help you prepare application programs to use sockets over SNA.

VTAM AnyNet Feature for V4R2: Guide to SNA over TCP/IP (SC31-6527): This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.

MPTN Architecture Publications

Networking Blueprint Executive Overview (GC31-7057)

Multiprotocol Transport Networking: Technical Overview (GC31-7073)

Multiprotocol Transport Networking: Formats (GC31-7074)

APPC Application Suite Feature for V4R2 for MVS/ESA

APPC Application Suite User's Guide (SC31-6532): This book documents the end-user interface (concepts, commands, and messages) for the AFTP, ANAME, and APING facilities of the APPC Application Suite for MVS/ESA. Although its primary audience is the end user, administrators and application programmers may also find it useful.

APPC Application Suite Administration (SC31-6533): This book contains the information that administrators need to configure the APPC Application Suite for MVS/ESA and to manage the APING, ANAME, AFTP, and A3270 servers.

APPC Application Suite Programming (SC31-6534): This book provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

Related Publications

ISPF Dialog Tag Language and Reference Guide (SC34-4267)

Index

Special Characters

#BATCH default class of service 557
 #BATCHSC default class of service 557
 #CONNECT default class of service 557
 #INTER default class of service 557
 #INTERSC default class of service 557

Numerics

1, type physical unit 149
 See also PU
 2, type physical unit 149
 See also PU
 2.1, non-native network connection 250, 490
 2.1, type peripheral node
 See PU
 2.1, type physical unit
 See PU
 3172 Interconnect Controller
 APPN connections 539
 connection network, defining a connection to 533
 link-attached NCP support 212
 multiple-domain connections
 CSMA/CD 802.3 network 274
 Fiber Distributed Data Interface network (FDDI) 274
 token-ring network 274
 NetView and 231
 single-domain connections
 CSMA/CD 802.3 network 231
 Fiber Distributed Data Interface network (FDDI) 231
 token-ring network 231
 tuning statistics 570
 VCNS and (MVS, VM) 195
 3272 cluster controllers 228
 See also cluster controllers
 3276 cluster controller 231
 See also cluster controllers
 3277 terminals 228
 3705, initial test routine for (VSE) 113
 3745 communication controller
 See communication controller
 8775 display terminal 231
 9000 series host processor
 peripheral node connections 234
 9000 series, Enterprise System host (VM, VSE)
 See integrated communication adapter (VM, VSE)

A

ACB (access method control block)
 opening, application program 161
 ACB macroinstruction
 APPLID operand 161
 PERSIST operand (MVS) 193
 ACBNAME operand on APPL definition statement 161
 accounting for cross-domain sessions 330
 accounting information, VSCS (VM)
 accounting record prefix (16 bytes) 647
 calculate average buffer size 647
 performance tuning assistance 647
 accounting record format (VSCS) (VM) 647
 activating an NCP
 activating adjacent link stations 371
 overview 371
 reactivating a failed NCP 374
 switched link, using 377
 VM restrictions on 376
 VSE restrictions on 376
 activating resources
 ADJSSCP tables 427
 application programs 349
 automatic activation of cross-subarea links 371
 cross-domain resource managers 427
 multiple-domain environment 427
 multiple-network environment 493
 NCP subarea node 371
 order of activation 349
 overview 348
 restrictions on channel-attachment node lines 427
 adaptive pacing, window size 334
 adaptive session pacing 334
 ADD command (IPL) (VSE) 115
 adding PUs and LUs dynamically 149, 291
 address space
 spanning multiple 195
 address subarea node
 address incompatibility 423
 addressing methods 422
 extended addressing compatibility 425
 extended network addressing 423
 path definition, subarea 421
 subarea addressing restrictions 426
 address translation 479
 address, element 306
 address, storage, translation 649
 addressing
 channel-attached SNA devices (VM)
 compatibility
 extended network and extended subarea 425
 non-extended and extended network 423

- addressing (*continued*)
 - LU, dependent 141
 - multiple-network (SNI) (MVS, VM) 473
 - overview 306
 - restrictions 307
 - structure types 306
 - subarea restrictions 426
 - translation of address 479
- ADDSESS operand on the BUILD definition statement 226
- adjacent CP
 - See control point (CP)
- adjacent link station (ALS) 218
- adjacent SSCP
 - lists for CDRSCs 400
 - multiple-domain environment 398
 - overview 398
 - performance 404
 - request routing 471
 - selecting with the session management exit 427
 - tables
 - alias name translation and 473
 - combined APPN and subarea network 512
 - default lists 467
 - defining 400, 467
 - dynamic CDRSCs and 472
 - dynamically defining 400
 - multiple-network environment 467
 - multiple-network sample 468
 - network-specific lists 467
 - requirements for 468
 - types of 467
- adjacent SSCP lists for CDRSCs 400
- ADJCP definition statement 526
- Advanced Peer-to-Peer Networking
 - See APPN
- alerts
 - VTAM to NPDA
 - NMVTLOG start option 133
- alias name translation
 - adjacent SSCP tables and 473
 - alias selection function (VTAM) 482
 - defining 485
 - LU 6.2 488
 - NetView 442
 - overview 483
- allocating private storage 195
- ALS (adjacent link station) 218
- ALSLIST operand
 - APPN connections 534
 - combined networks 512
 - independent LUs 222
- ALSREQ start option 223, 226
- analysis tool for VIT
 - data sets 22
 - installing 35, 96
- ANS operand on PU definition statement 252
- AnyNet feature 697
- APBUF buffer pool (MVS)
 - number of buffers per page 590
 - overview 583
- APF (authorized program facility) 163
- API, cross-memory (MVS) 195
- APPC operand on APPL definition statement 196
- APPCCMD macroinstruction
 - CONFTXT operand 173
 - requirement for LU 6.2 196
- APPL definition statement
 - ACBNAME operand 161
 - APPC operand 196
 - ATNLOSS operand 201
 - AUTH operand 162
 - AUTODMP operand 203
 - AUTOIPL operand 203
 - AUTOSES operand 198
 - DDRAINL operand 198
 - description 138
 - DLOGMODE operand 173
 - DMINWNL operand 198
 - DMINWNR operand 198
 - DRESPL operand 198
 - DSESLIM operand 198
 - EAS operand 578
 - ENCR operand (MVS) 169
 - HAVAIL operand (MVS) 174
 - LIMQSINT operand 202
 - LIMRES operand 202
 - MAXPVT operand 195
 - MODETAB operand 173
 - PARSESS operand 163, 481
 - sample 138
 - SECACPT operand 200
 - SYNCLVL operand 201
 - VCNS operand (MVS, VM) 195
- application password 659
- application program
 - ACBNAME operand 161
 - acquiring sessions 163
 - activating 349
 - APPLID operand 161
 - ATTN exit 201
 - authorized path 163
 - communication network management (CNM) 360
 - confidential data 173
 - controlling 323
 - cross-memory API (MVS) 195
 - data transfer, privileged (MVS) 163
 - defining 138
 - dial number digits, overriding 163
 - draining conversations 198
 - encryption (MVS) 168
 - extended recovery facility (XRF) 174

- application program (*continued*)
 - failed sessions, reconnecting to 193
 - generic resources 182
 - impact of outages, reducing 174
 - keeping active (VSE) 119
 - logon mode 173
 - logon mode parameters 173
 - logon mode table entries, create 173
 - logon request validation 162
 - LU 6.2 196
 - LU 6.2 and XRF 193
 - LU 6.2 security 199
 - LU 6.2 session 196
 - LU 6.2 sync point service 201
 - naming 161
 - network management capability 358
 - overview 125
 - pacing 337
 - parallel sessions 163, 481
 - passing logon request 162
 - private storage, allocating 195
 - program operator (PPO and SPO) 358
 - sample definition 138
 - security facilities 168
 - security protocol, conversation 199
 - spanning multiple address space 195
 - start-stop devices, communicating with 196
 - starting
 - name of application and 162
 - overview 172
 - TSO (MVS) 611
 - USERVAR 175
 - validate logon 162
 - VSCS (VM) 627
 - XRF (extended recovery facility) 174
- application-supplied operands for dial or token ring 163
- APPLID operand on ACB macroinstruction 161
- APPN
 - addressing
 - overview 19
 - ALSLIST operand on CDRSC definition statement 534
 - bisynchronous (BSC) 3270 sessions 502
 - border node 545
 - connection types 691
 - BSC 3270 sessions 502
 - central directory server
 - benefits of 553
 - overview 7
 - resource registration 554
 - search requests 554
 - using to improve efficiency 553
 - checkpoint data sets 26
 - class of service (COS)
 - choosing a route 558
 - defaults 557
- APPN (*continued*)
 - class of service (COS) (*continued*)
 - description 556
 - installing the default file 25
 - logon mode table and 558
 - overview 17
 - combined with subarea network 497
 - composite network node
 - overview 9
 - connection network 528
 - connections
 - 3172 interconnect controller 539
 - assigning NCP addresses 535
 - definitions of 533
 - leased lines 537
 - overview 527
 - requirements 534
 - sample type 2.1 channel 536
 - session control block usage 535
 - setting maximum number of sessions 535
 - type 2.1 channel 497, 535
 - control point (CP)
 - CP name 128, 526
 - overview 5
 - converting LEN connections 497, 498
 - CP name for VTAM 128
 - CP-CP sessions
 - overview 12
 - CPNAME operand on PU definition statement 534
 - dependent LUs 502
 - directed search 551
 - directory services
 - controlling the size of 555
 - end node
 - overview 6
 - extended subnetwork boundaries 548
 - generic resources 182
 - host-to-host channel connection 536
 - IDBLK operand on PU definition statement 534
 - IDNUM operand on PU definition statement 534
 - implementing a network 525
 - interchange node
 - coding 498
 - combined APPN and subarea network 498
 - overview 8
 - LEN connections, converting 497, 498
 - locating nodes 501, 550
 - migrating from subarea 497
 - migration data host 499
 - overview 8
 - multiple network connectivity 545
 - network accessible units 10
 - network node
 - overview 5
 - network node server
 - defining list 544
 - overview 7

APPN (*continued*)

- network node server (*continued*)
 - replacing list 545
 - resource registration 554
 - sample list 545
- nodes
 - adding and moving 522
 - adjacent 526
 - connecting to VTAM 527
 - types of 4
- peripheral subnetwork boundary 548
- resource registration 554
- route calculation
 - overview 18
- route selection
 - overview 18, 558
- routing
 - class of service and 558
 - directory services 550
 - forcing 685
 - improving efficiency 553
 - overview 16
 - route selection 18, 558
 - topology database 16
 - types of searches 551
- searches
 - broadcast search 552
 - directed search 551
 - improving efficiency 553
- sessions in a combined network 490, 512
- start options
 - APPNCOS 549, 558
 - BN 549
 - BNDYN 549
 - BNORD 549
 - CDSERV 554
 - CONNTYPE 497
 - CPCP 525
 - DIRSIZE 555
 - DIRTIME 555
 - DYNADJCP 526
 - INITDB 556
 - NODETYPE 498, 499, 525
 - RESUSAGE 559
 - SNVC 549
 - VERIFYCP 200
 - VRTG 517
 - VRTGCPCP 517
- XID operand on PU definition statement 534
- APPNCOS start option 549, 558
- associated LU table 157
- assuming resource ownership 432
- ATCCON (configuration list) 134
- ATCSTR00 (start option list) 127, 129, 139
- ATNLOSS operand on APPL definition statement 201

- ATTN exit 201
- AUTH operand on APPL definition statement 162
- AUTHLEN start option and operand 227
- authorized path 163
- authorized program facility (APF) 163
- authorized transmission priority for LEN
 - connections 227
- authorizing cross-domain sessions 329
- authorizing VSCS in GCS (VM) 628
- AUTODMP operand on APPL definition statement 203
- AUTOIPL operand on APPL definition statement 203
- automatic logon
 - changing an LU's automatic logon specification 325
 - coding 158
 - controlling 329
 - defined 158
 - fails, warning 180
 - LOGAPPL operand 158, 323
 - multiple-network 491
 - operator-controlled 325
- AUTORTRY start option 326
- AUTOSOS operand on APPL definition statement 198
- AUTOSYN operand on PCCU definition statement 204
- AUTOTI start option 326
- AUXADDR operand on the BUILD definition
 - statement 226

B

- backup
 - NCP owner 432
 - sessions, extended recovery facility and 174
- backup start option list (LISTBKUP) 129, 130
- binary synchronous (BSC) connection
 - and APPN 502
 - NCP-attached 259
 - VTAM-attached 248
- BIND, extended 218
- BN start option 549
- BNDYN start option 549
- BNORD start option 549
- border node 545
 - connection types 691
- boundary function-based transmission groups 527
- broadcast search (APPN)
 - central directory server and 552
 - limiting 513, 553
 - originating 552
 - overview 552
 - when used 552
- BSBUF buffer pool
 - overview 583
- BSC (binary synchronous) connection
 - and APPN 502
 - NCP-attached 259
 - VTAM-attached 248

- buffer pools
 - allocating storage 585
 - dynamic expansion
 - basic allocation and 585
 - guidelines for 587
 - operation of 586
 - virtual storage and 591
 - monitoring 567
 - number of buffers per page 590
 - overview 566
 - types of 583
- BUILD definition statement
 - description 207
 - LOADLIB operand 205
 - NEWNAME operand 205
 - sample, channel-attached 205, 208

C

- CACHETI start option 549
- call security verification 271
- canceling VTAM 358
- casual connection
 - alternate application program 174
 - APPL definition statement, HAVAIL operand 174
 - defining 408
 - LU 6.2 in an XRF Complex 193
 - restrictions on casual connection 413
 - type 2.1 peripheral node 407
 - USERVAR map 175
- CCS (console communication services) 628
- CDRM (cross-domain resource manager)
 - activating 427
 - coding of 393
 - deactivating 428
 - description of 393
 - DISJOINT operand 513
 - dynamic definition of cross-domain resources 394
 - external CDRM 393
 - gateway VTAM, for (MVS, VM) 455
 - host CDRM 393
 - multiple domains 393
 - sample of 393
- CDRSC (cross-domain resource)
 - adjacent SSCP (ADJSSCP) table, defining 467
 - adjacent SSCP lists for CDRSCs 400
 - CDRM ownership
 - changing by operator 406
 - changing dynamically 406
 - verifying 406
 - combined APPN and subarea network 508
 - defining 395, 508
 - definition statement
 - VFYOWNER operand 406
 - description 467
 - dynamic 465
 - CDRSC (cross-domain resource) (*continued*)
 - locating
 - adjacent SSCP table 398
 - ADJCDRM definition statement 401
 - CDRSC routed to each SSCP 399
 - cross-network 467
 - multiple-network configuration 468
 - multiple-network environment 458
 - performance improving 404
 - predefined 462
 - SSCPDYN start option 404
 - SSCPORD start option 404
 - CDRSCTI start option 502, 513
 - CDSERVR start option (APPN) 554
 - central directory server (APPN)
 - benefits of 553
 - broadcast search and 552
 - defining 554
 - overview 7
 - resource registration 554
 - search requests 554
 - using to improve efficiency 553
 - change-number-of-sessions
 - contention loser and 198
 - modifying parameters 197
 - operating commands for 199
 - overview 197
- channel
 - connections, VTAM-to-peripheral node 228
 - connections, VTAM-to-VTAM 262
 - device name
 - non-SNA devices, for 228
 - SNA devices, for 229
 - enterprise systems connection (ESCON) 230
 - multipath channel (MPC) for hosts 266
 - sample APPN 536
 - type 2.1 APPN 535
- channel unit address
 - ICA token-ring (VM, VSE) 240
 - NCP, channel-attached 208
- channel-attached
 - APPN type 2.1 535
 - between host 262
 - devices, dynamic definition of 32
 - I/O performance with nodes 569
 - loop-adapter attachment 231
 - MVS/CP devices 147, 287
 - NCP 207
 - non-SNA devices 228
 - single-domain 228
 - SNA cluster controllers
 - defining statically 229
 - ES/9000 series processors and (VM, VSE) 229
 - terminals 650
 - type 2.1 140

Index

- channel-link
 - NCP 209
 - parallel transmission group and 209
- channel-to-channel adapter connections
 - defining 262
 - ES/9000 series (VM, VSE) 278
 - tuning I/O for 571
- checkpointing APPN data sets 26
- checkpointing TRS and directory databases 555
- CICS
 - extended recovery facility (XRF) 174
- class of service (COS)
 - APPN
 - choosing a route 558
 - defaults 557
 - description 556
 - installing the default file 25
 - logon mode table and 558
 - overview 17, 556
 - weight computation 559
 - combined APPN and subarea network 520
 - COSAPPN file 25
 - database (APPN) 17
 - defaults, IBM-specified 416
 - logon mode table 173
 - multiple-network 477
 - name 484
 - substituting parameters 416
 - table 415
 - unnamed entry 416
- cluster controllers
 - non-SNA, defining 228
- SNA
 - 3276 231
 - defining statically 229
 - I/O performance to 569
 - sample tuning statistics report 570
- CLUSTER definition statement (VM, VSE)
 - BSC single-domain connection 248
- CMC (communication management configuration)
 - overview 4, 145
 - problem determination 287
- CMPMIPS start option 164
- CNM (communication network management)
 - application program 359
 - application program, defined 359
 - filtering session awareness data 360
 - NetView program and 443
 - procedure-related identifier (PRID) 360
 - routing table 359
 - routing unsolicited requests 359
- CNOS
 - contention loser and 198
 - modifying parameters 197
 - operating commands for 199
 - overview 197
- coattailing
 - channel-to-channel adapter
 - DELAY operand 603
 - I/O buffer size 602
 - virtual route (VR) window size 604
 - DELAY operand 596
 - determining amount of 577
 - inbound 596
 - MAXBFRU operand 597
 - maximizing 595
 - outbound 595
 - overview 567
 - UNITSZ value 597
- coded data (MVS)
 - CDRM key 665
 - cross-domain cryptography 665
 - filing secondary logic unit (SLU) keys 663
 - single-domain encryption 663
- coding concepts 136
- coding resource definition statements
 - overview 136
 - sift-down 137
- combined APPN and subarea network 497
- command set library
 - installing 48
- command, compiling with ISPF 40, 100
- commands, operator
 - See operator commands*
- common service area (CSA)
 - buffer pool waste 568
 - common service area 24-bit (CSA24) (MVS, VM) 579
 - CSA limit 578
 - monitoring 567
- communication controller
 - adapter type for NTRI 257
 - channel-enable switch 378
 - dumping of 385
 - enabling channels for activation 378
 - I/O performance with 569
 - links between
 - activating 381
 - deactivating 375
 - loading or purging the disk 370
 - remote program load (RPL) facility 378
 - sample tuning statistics report 570
 - tuning I/O for 570
- communication management configuration (CMC)
 - overview 4, 145
 - problem determination 287
- communication network management (CNM)
 - application program 359
 - application program, defined 359
 - filtering session awareness data 360
 - NetView program and 443
 - procedure-related identifier (PRID) 360

- communication network management (CNM) (*continued*)
 - routing table 359
 - routing unsolicited requests 359
- communication scanner processor (CSP) 31
- composite network node
 - overview 9
- compression of data (MVS) 163
- concatenating
 - target data sets for dump and trace tools 36, 96
 - USER1.AUTO.VTAMLST 28
- confidential data 173
- CONFIG operand on START command 134
- configuration list
 - advantages of 127
 - CONFIG operand 134
 - for gateway VTAM (MVS, VM) 442
 - overview 125
 - sample 134
- configuration of I/O devices, dynamic (MVS)
 - command list programs 673
 - customization 174, 290
 - default naming convention 289
 - installation and preparation 27, 287
 - overview 147, 287
- configuration restart (MVS, VSE)
 - defined 342
 - NODELST data set 342
 - WARM start 343
- configuration restart file, VSAM (MVS, VSE)
 - associating with major nodes 347
 - CONFGDS operand 347
 - CONFGPW operand 347
 - MVS 347
 - MVS installation 27
 - VSE 111
 - VSE installation 111
- CONFTXT operand on APPCCMD
 - macroinstruction 173
- congestion, route 311
- connecting networks 443
- connection networks (APPN) 528
- connections, leased
 - See* nonswitched SDLC connection
- connections, switched
 - See* switched SDLC connection
- CONNTYPE
 - operand on PU definition statement
 - overview 534
 - parallel transmission groups and 528
 - start option 497, 534
- console communication services (CCS) 628
- consoles, multiple operator (MVS) 353
- contention loser
 - CNOS and 198
- continuous executing transfer interface (CETI)
 - group 241
- control point (CP)
 - adjacent
 - ADJCP definition statement 526
 - DYNADJCP start option 526
 - predefinition of 526
 - dynamic creation of adjacent CPs 526
 - overview 5
- conversations, LU 6.2
 - draining 198
 - overview 197
- COS (class of service)
 - APPN
 - choosing a route 558
 - defaults 557
 - description 556
 - installing the default file 25
 - logon mode table and 558
 - overview 17, 556
 - weight computation 559
 - combined APPN and subarea network 520
 - COSAPPN file 25
 - database (APPN) 17
 - defaults, IBM-specified 416
 - logon mode table 173
 - multiple-network 477
 - name 484
 - substituting parameters 416
 - table 415
 - unnamed entry 416
- CP (control point)
 - adjacent
 - ADJCP definition statement 526
 - DYNADJCP start option 526
 - predefinition of 526
 - dynamic creation of adjacent CPs 526
 - overview 5
- CP name for VTAM 128
- CP system name table (DMKSNT) 52
- CP-CP sessions (APPN)
 - establishing between VTAM nodes 526
 - overview 12
- CPCP
 - operand on PU definition statement 533
 - start option 525
- CPNAME operand on PU definition statement
 - type 2.1 PU and 139
 - VTAM switched SDLC connection 237, 253
- CPSVCMG default class of service 557
- CPSVRMGR pipe 503
- CPSVRMGR session 502
- cross-domain resource (CDRSC)
 - adjacent SSCP (ADJSSCP) table, defining 467
 - adjacent SSCP lists for CDRSCs 400
 - CDRM ownership
 - changing by operator 406

- cross-domain resource (CDRSC) (*continued*)
 - CDRM ownership (*continued*)
 - changing dynamically 406
 - verifying 406
 - combined APPN and subarea network 508
 - defining 395, 508
 - definition statement
 - VFYOWNER operand 406
 - description 467
 - dynamic 465
 - locating
 - adjacent SSCP table 398
 - ADJCDRM definition statement 401
 - CDRSC routed to each SSCP 399
 - cross-network 467
 - multiple-network configuration 468
 - multiple-network environment 458
 - performance improving 404
 - predefined 462
 - SSCPDYN start option 404
 - SSCPORD start option 404
 - cross-domain resource control
 - CDRSC definition statement 398
 - controlled by another VTAM domain 397
 - cross-domain resource major node 397
 - dynamic adjacent SSCP table 400
 - locating cross-domain resources 398
 - network resources for XRF 398
 - USERVAR with XRF 398
 - cross-domain resource manager (CDRM)
 - activating 427
 - coding of 393
 - deactivating 428
 - description of 393
 - DISJOINT operand 513
 - dynamic definition of cross-domain resources 394
 - external CDRM 393
 - gateway VTAM, for (MVS, VM) 455
 - host CDRM 393
 - multiple domains 393
 - sample of 393
 - cross-memory API (MVS) 195
 - cross-network routing
 - communication between network 474
 - COS tables 479
 - extended subarea address gateway 474
 - extended subarea addressing 474
 - gateway NCP 474
 - gateway VTAM (MVS, VM) 474
 - virtual route, default 479
 - CRPLBUF buffer pool
 - guidelines for 589
 - overview 584
 - cryptographic capability, changing (MVS) 671
 - cryptographic key (MVS)
 - CDRM key 665
 - cryptographic key (MVS) (*continued*)
 - cross-domain cryptography 665
 - filing secondary logic unit (SLU) keys 663
 - single-domain encryption 663
 - cryptography (MVS)
 - application program 168
 - APPN and cryptography 171
 - end-to-end cryptography 170
 - extended recovery facility (XRF) 175
 - host-by-host cryptography 170
 - sample
 - multiple-domain 170
 - single-domain 169
 - CSA (common service area)
 - buffer pool waste 568
 - common service area 24-bit (CSA24) (MVS, VM) 579
 - CSA limit 578
 - monitoring 567
 - CSA24 start option (MVS, VM) 579
 - CSALIMIT start option 578
 - CSMA/CD 802.3 network 231, 274
 - CSP (communication scanner processor) 31
 - CTCA
 - defining 262
 - ES/9000 series (VM, VSE) 278
 - tuning I/O for 571
 - CUADDR operand
 - ICA token-ring (VM, VSE) 240
 - NCP, channel-attached 208
 - CWALL threshold condition (NCP) 569
- ## D
- data compression (MVS) 163
 - data control, subarea to network 310
 - data hosts
 - overview 4
 - data sets (MVS) 21
 - data sets, VTAM- and NCP-specific 25
 - data spaces 115
 - data streams, modifying (VSCS) (VM) 632
 - data transmission shutdown 311
 - DDRAINL operand on APPL definition statement 198
 - deactivating resources
 - automatic deactivation of cross-subarea links 355
 - cross-domain resource managers 428
 - forced deactivation 356
 - forced reactivation 356
 - immediate deactivation 356
 - multiple-domain environment, in a 428
 - NCP subarea node 381
 - normal deactivation 355
 - order of deactivation 355
 - overview 355

- DELAY operand on LINE definition statement
 - CTC coattailing and 603
 - inbound coattailing and 596
 - NCP coattailing and 598
- delayed activation of logical links
 - multiple-domain, in a 430
 - single-domain, in a 375
- delayed disconnection 255
- deleting PUs and LUs dynamically 149, 291
- dependent logical unit (LU)
 - See logical unit
- dependent LU requester 502
- dependent LU server 502
- dependent LUs in an APPN network 502
- DIAG98 option (VM) 649
- diagnosis
 - alerts to NPDA 133
 - module name, viewing 134
 - NMVTLOG start option 133
 - SDLC statistical MDRs 134
 - start options 133
 - time interval notification 134
 - trace facility 134
- dial connections
 - dial-in
 - VTAM switched SDLC 236, 253
 - dial-out
 - auto (to NCP) 215
 - manual (to NCP) 216
 - VTAM switched SDLC 254
- dial number digits
 - overriding 163
- DIALNO operand on PATH definition statement
 - NTRI, for 258
 - V.25 bis (VM, VSE) 216
- direct call, X.21
 - PATH definition statement 238
- directed search (APPN)
 - originating 551
 - overview 551
 - when used 551
- directory entries, creating (VM)
 - 5654010A virtual machine
 - minidisk address, defaults 55
 - required minidisk table 55
- directory services (APPN)
 - checkpointing 555
 - controlling the size of the database 555
 - database 550
 - improving performance 555
 - learning about resources 550
 - overview 550
- DIRSIZE start option (APPN) 555
- DIRTIME start option (APPN) 555
- DISCNT (disconnect facility) 255
- DISCNTIM start option 256
- disconnect facility (DISCNT) 255
- disconnection, delayed (DISCNT=DELAY) 256
- discontiguous domains 432
- DISJOINT operand on CDRM definition statement 513
- DISPLAY BFRUSE operator command 567
- DISPLAY STORUSE pools 579
- DLCADDR operand on PATH definition statement
 - dependent LU requester, defining 504
 - NTRI, for 258
- DLOGMODE operand on APPL definition statement 173
- DLUR 502
- DLURNAME operand on PATH definition statement 504
- DLUS 502
- DMINWNL operand on APPL definition statement 198
- DMINWNR operand on APPL definition statement 198
- DMKSNT (CP system name table) (VM) 52
- DMPFREQ (VSCS) (VM) 635
- DMPSYSC (VSCS) (VM) 635
- DMPTIME (VSCS) (VM) 636
- DRESPL operand on APPL definition statement 198
- DSESLIM operand on APPL definition statement 198
- DTE-to-DTE connections (VM, VSE)
 - multiple domain 282
 - single-domain 247
- DTIGEN macroinstruction (VM)
 - controlling storage management
 - STCHKTM option 633
 - STRELTM option 633
 - personalized 639
 - SCIP recovery 638
 - VSCS dumps, controlling 635
 - VSCS initialization parameters 639
 - VSCS performance, controlling 636
- dump data set (CSP) 31
- dump data set (MOSS) 31
- dump formatter panels, verifying 42
- dumps
 - analysis
 - data sets 22
 - installing 35
 - formatter panels, verifying 42
 - NCP
 - dynamic 382
 - performance, improving 133
 - selecting a dump file 385
 - selecting the disposition of a dump 385
 - static (destructive) 382
 - VSCS, controlling (VM) 635
 - VSE files for 113
- DYNADJCP start option 526
- dynamic definition of CDRSCs 395
- dynamic definition of independent LUs 221

- dynamic definition of PUs 143
 - dynamic expansion (of buffer pools)
 - See buffer pools
 - dynamic I/O configuration (MVS)
 - command list programs 673
 - customization 174, 290
 - default naming convention 289
 - installation and preparation 27, 287
 - overview 147, 287
 - dynamic network identification 451
 - dynamic path update 313
 - dynamic reconfiguration
 - effect on WARM start 347
 - frame relay lines, PUs on 298
 - implicit dynamic reconfiguration 300
 - overview 149, 291
 - samples 301
 - VARY ACT, UPDATE technique 292
 - VARY DRDS technique 295
 - dynamic resource definition (model name) 157
 - dynamic selection of session connections 225
 - dynamic switched definitions 143, 260
 - dynamic switched major node 144
 - dynamic table replacement 354
 - DYNLU start option 221, 526
 - DYNPU operand 143
- ## E
- EAS (estimated number of active sessions)
 - APPL definitions 578
 - improving session establishment performance 227
 - LFBUF and 589
 - LU definitions 578
 - TSO, specifying for (MVS) 612
 - element address 306
 - ENA (extended network addressing)
 - addressing compatibility
 - ERLIMIT operand 309
 - extended subarea 309
 - network configuration 309
 - SALIMIT operand 309
 - addressing restrictions 307
 - compatibility, non-extended address 307
 - element address 306
 - ENCR operand (MVS)
 - APPL definition statement 169
 - LU definition statement 169
 - encryption facility (MVS)
 - cryptographic keys, defining 663
 - description 168
 - extended recovery facility (XRF) 175
 - sample
 - multiple-domain 170
 - single-domain 169
 - encryption of data (MVS)
 - application program 168
 - APPN and cryptography 171
 - CDRM key 665
 - cross-domain cryptography 665
 - end-to-end cryptography 170
 - extended recovery facility (XRF) 175
 - filing secondary logic unit (SLU) keys 663
 - host-by-host cryptography 170
 - sample
 - multiple-domain 170
 - single-domain 169
 - single-domain encryption 663
 - encryption of data, changing (MVS) 671
 - end nodes (APPN)
 - coding 499
 - combined APPN and subarea network 499
 - overview 6
 - resource registration 554
 - ENHADDR start option 306
 - enterprise system connection channel 230
 - environment, multiple-network
 - activating resources 493
 - combined APPN and subarea network 513
 - disjointed networks 513
 - network management with NetView 493
 - terminating sessions 493
 - error recovery activation 252
 - ES/9000 host processor connections (VM, VSE)
 - See integrated communication adapter (VM, VSE)
 - ES/9221 host processor connections (VM, VSE)
 - See integrated communication adapter (VM, VSE)
 - ESCON 230
 - estimated number of active sessions (EAS)
 - APPL definitions 578
 - improving session establishment performance 227
 - LFBUF and 589
 - LU definitions 578
 - TSO, specifying for (MVS) 612
 - exchange of ID processing
 - type 2.1 PUs 140
 - VTAM switched SDLC connection 254
 - exit routine
 - session accounting 329
 - session authorization 329
 - warning 329
 - exit routine, session management
 - See session management exit routine (SME)
 - exit routines, data manipulation (VSCS) (VM) 632
 - explicit route 151
 - extended BIND 218
 - extended network addressing 480
 - extended network addressing (ENA)
 - addressing compatibility
 - ERLIMIT operand 309
 - extended subarea 309
 - network configuration 309

extended network addressing (ENA) (*continued*)
 addressing compatibility (*continued*)
 SALIMIT operand 309
 addressing restrictions 307
 compatibility, non-extended address 307
 element address 306
 extended recovery facility (XRF) (MVS)
 cryptography and 175
 description 174
 extended subnetwork boundaries 548
 external communication adapter (XCA)
 APPN connections 539
 connection network, defining a connection to 533
 link-attached NCP support 212
 multiple-domain connections
 CSMA/CD 802.3 network 274
 Fiber Distributed Data Interface network (FDDI) 274
 token-ring network 274
 NetView and 231
 single-domain connections
 CSMA/CD 802.3 network 231
 Fiber Distributed Data Interface network (FDDI) 231
 token-ring network 231
 tuning statistics 570
 VCNS and (MVS, VM) 195

F

failure, warning on initiation 180
 FDDI 231, 274
 FFST 29, 104
 Fiber Distributed Data Interface 231, 274
 filters
 NetView filter 360
 session awareness data filter table 360
 session monitor filter 360
 VTAM filter 360
 first failure support technology 29, 104
 flow control, data 159
 flow control, subarea to network 310
 forced reactivation 356
 frame relay lines, dynamic reconfiguration 298

G

gateway NCP
 backup 451
 GWNAU definition statement 446
 native network
 BUILD definition statement 447
 GWNAU definition statement 448
 HOST definition statement 449
 NCP PATH definition statements 450
 PU definition statement 450

gateway NCP (*continued*)
 non-native network
 GWNAU definition statement 450
 NETWORK definition statement 450
 gateway NCP backup 451
 gateway VTAM (MVS, VM)
 CDRM definition statement 456
 defining 445
 GWPATH definition statement 456
 NETWORK definition statement 455
 GCS (group control system) (VM) 84
 GCS (group control system), (VM) 629
 GCS ACCESS command
 start option list and 131
 VTAMLST file 137
 GCS and VSCS environment (VM) 628, 629
 generic resources 182
 GID operand on PATH definition statement 254
 group control system (GCS) (VM) 629
 group control system (GCS), (VM) 84
 GROUP definition statement
 ICA connections (VM, VSE)
 nonswitched SDLC 234
 PAUSE operand 260
 POLIMIT operand 260
 PU operand 259
 SHOLD operand 237
 X.25 single-domain connection (VM, VSE) 244
 GWSSCP start option 442

H

HALT operator command 356
 halting VTAM 356
 Hardware Configuration Definition (HCD) (MVS) 147
 HAVAIL operand on APPL definition statement (MVS) 174
 HCD (Hardware Configuration Definition) (MVS) 147
 help panels, compiling with ISPF 40, 100
 HOLD operand
 overriding
 VTAM peripheral nodes 256
 session termination
 VTAM peripheral nodes 256
 HOST definition statement
 MAXDATA operand 206
 sample 206
 UNITSZ operand 597
 host physical unit name 128
 host-to-host channel connection
 APPN, defining 536
 host, restart 433
 HOSTPU start option
 changing 128
 ISTPUS major node 137
 multiple-network 441

HOT I/O detection and termination 592

I

I/O (input/output) operations

analyzing tuning statistics 570

information gathering 570

IBM Command Tree/2

installing 48

ICA (VM, VSE)

See integrated communication adapter (VM, VSE)

ICSF (Integrated Cryptographic Service Facility)

(MVS) 168

IDBLK operand on PU definition statement

type 2.1 PU 139

VTAM switched SDLC connection 237, 253

identifying host, single-domain 127

IDNUM operand on PU definition statement

type 2.1 PU 139

VTAM switched SDLC connection 237, 253

ILU (independent LU)

adjacent link station 218

characteristics 218

defining 220

description 218

dynamic definition of 221

dynamic selection of sessions 225

extended BIND 218

multiple connections 224

NCP connections 226

overview 11, 140, 218

restrictions 227

sessions in a combined network 512

implicit dynamic reconfiguration 300

IMS

extended recovery facility (XRF) 174

independent LU (ILU)

adjacent link station 218

characteristics 218

defining 220

description 218

dynamic definition of 221

dynamic selection of sessions 225

extended BIND 218

multiple connections 224

NCP connections 226

overview 11, 140, 218

restrictions 227

sessions in a combined network 512

Information Management System

extended recovery facility (XRF) 174

initial program load procedures (VSE) 114

initial status

overview 137

warm start and 137

initial test routine for IBM 3705 (VSE) 113

initiation failure 180

initiation, automatic 323

input/output (I/O) operations

analyzing tuning statistics 570

information gathering 570

installing VTAM under MVS

data sets for installation

configuration restart data set 27

NODELST data set 29

O/S data sets used by VTAM 21

SYS1.VTAMLIB (required) 25

SYS1.VTAMLST (required) 24

VTAM- and NCP-specific data sets 24

defining VTAM to MVS

channel-attached devices, adding 32

DATAMGT macroinstruction 32

IODEVICE macroinstruction 32

name system data sets 31

routing codes changes 32

SECONSLE macroinstruction 32

installing, using distribution tape 34

NCP-specific data sets

NCP dump data set 30

NCP load library 29

start procedure VTAM, defining

name of start procedure 34

sample of start procedure JCL 34

start procedure name 34

SYS1.PROCLIB 34

installing VTAM under VM

preparing your VM system

CP system name table (DMKSNT) 52

generating group control system (GCS) 84

servicing VTAM 86

setting up VM directory entries

5654010A virtual machine 55

VTAM virtual machine 52

installing VTAM under VSE 107

integrated communication adapter (VM, VSE)

BSC connection 248

channel-attached NCP connection 278

channel-to-channel adapter 278

connecting hosts with 278

nonswitched SDLC connections

multiple-domains 278

peripheral nodes (VTAM) 234

single-domain connections 234

switched SDLC connection

multiple-domains 279

peripheral nodes (VTAM) 235

token-ring connection

multiple subsystems 242

multiple-domains 283

peripheral nodes (VTAM) 239

X.25 connection

DTE-to-DTE single-domain connection 247

integrated communication adapter (VM, VSE) (*continued*)

X.25 connection (*continued*)

multiple-domains 280

peripheral nodes (VTAM) 243

Integrated Cryptographic Service Facility (ICSF) (MVS) 168

intensive mode recording (IMR) 379

inter-user communication vehicle (IUCV) 628

Interactive System Product Facility (ISPF)

dialog tag language utility 39, 99

installing 38, 98

interface, customizing 46, 102

interchange node (APPN)

coding 498

combined APPN and subarea network 498

overview 8

interconnected networks, sample 462

intermediate routing node 421

International Standards Organization

See ISO standards

interpret table

defined 156

defining for TSO (MVS) 620

LOGTAB operand 156, 251

nonswitched SDLC connection 251

sample 620

IOBUF buffer pool

application data transfer 591

coattailing guidelines 601

MAXBFRU operand and 597

overview 584

IOINT start option 134

IOPURGE start option 392

IPCS CLISTs 22, 28

IPCS interface, customizing 45

IPL and dump contention 435

IPL procedures (VSE) 114

IPL, timed 366

ISO standards

NETID 128

SSCPID 127

SSCPNAME 128

ISPF (Interactive System Product Facility)

dialog tag language utility 39, 99

installing 38, 98

interface, customizing 46, 102

ISTAPNPU generic representation 223, 534

ISTAPPLS VTAMLST 627

ISTCDRDY major node 137, 396

ISTCOSDF 323, 324, 415

ISTDEFIN command list 147, 288, 677

ISTDINFO command list 147, 288, 673

ISTDSWMN major node

initial status and 137

ISTINCLM 415

ISTMSFLD default message flooding prevention table 351

ISTPDILU major node

initial status and 137

ISTPUS major node

non-SNA devices and 228

overview 137

ISTVTCOS 415

ISTVTMEU translation file 682, 684

IUCV (inter-user communication vehicle) 628

J

JCL for the logon manager (MVS) 654

K

keylists, compiling with ISPF 40, 100

L

LAN (local area network)

3172 Interconnect Controller

single-domain connections 231

connections, sample 241

CSMA/CD 802.3 231

FDDI 231

token-ring

ICA (ES/9000) single-domain (VM, VSE) 239

XCA (3172) single-domain 231

type 2.1 PU and 140

XCA (3172)

single-domain connections 231

LANG operand (MVS) 316

LANGTAB operand 316

LBUILD definition statement

non-SNA devices, defining 228

sample 229

leased lines

APPN 537

sample APPN 538

LEN node

adjacent link station list 512

combined APPN and subarea network 497

migrating to APPN nodes 498

migrating to LEN nodes 497

overview 9

LFBUF buffer pool

application data transfer 591

EAS values and 589

guidelines for 589

overview 584

TSO and (MVS) 589

licensing agreement xxvii

- limited resources, LU 6.2
 - description 201
 - non-LU 6.2 sessions and 201
- LIMQSINT operand on APPL definition statement 202
- LIMRES operand on APPL definition statement 202
- line control operand (LNCTL)
 - channel-attached NCP 264
 - logon manager 655
 - token-ring network 239, 283
- LINE definition statement
 - BSC single-domain connection 248
 - DELAY operand
 - CTC coattailing and 603
 - inbound coattailing and 596
 - NCP coattailing and 598
 - ICA connections (VM, VSE)
 - nonswitched SDLC 234
 - switched SDLC 236
 - MODE operand 140
 - X.25 single-domain connection 244
 - XMITDLY operand 249
- lines
 - errors, recording and sending to VTAM 379
 - speed
 - reducing buffer use and 595
- link stations
 - overview 429
 - PRI or SEC indicator 140
 - primary or secondary, determining 140
 - sample of activating 429
 - specifying during NCP activation 372
- link-attached NCP
 - activating 371
 - LINE definition statement 212
 - overview 212
 - PU definition statement 212
 - recovery operations for 212
 - type 2.1 PUs, attached to 249
 - X.21 lines (VSE) 143, 259
 - XCA (3172) and 212
- link-level role negotiation
 - type 2.1 PU 140
- links
 - activating 429
 - deactivating 429
 - overview 429
- LIST option of START command 130
- LISTBKUP (backup start option list) 129, 130
- LNCTL operand on the GROUP definition statement
 - channel-attached NCP 264
 - logon manager 655
 - token-ring network 239, 283
- load library, NCP 29
- load module, NCP
 - identifying 205
 - multiple, maintaining 206
- load module, NCP (*continued*)
 - time stamps 204
 - verifying 204
- LOAD operand 362
- load station
 - IPL-capability of NCP and communication links 364
 - LOADSTA operand 364
 - selecting 364
- LOADER output (VM)
- loading a communication controller disk 370
- loading an NCP
 - performance, improving 133
 - reloading, automatic (scheduled) 366
 - switched link, using 432
- local area network (LAN)
 - 3172 Interconnect Controller
 - single-domain connections 231
 - connections, sample 241
 - CSMA/CD 802.3 231
 - FDDI 231
 - token-ring
 - ICA (ES/9000) single-domain (VM, VSE) 239
 - XCA (3172) single-domain 231
 - type 2.1 PU and 140
 - XCA (3172)
 - single-domain connections 231
- LOCAL definition statement 229
- local non-SNA devices
 - See* non-SNA devices
- local SNA devices
 - See* SNA devices
- local topology database (APPN)
 - See* topology database (APPN), local
- locating resources
 - APPN network 14
 - subarea network 14
- logical unit
 - addressing
 - dependent LU 141
 - dependent
 - addressing 141
 - combined subarea and APPN network 502
 - defining 141
 - overview 10
 - ownership of 141
 - secondary and 141
 - session establishment 141
 - description 139
 - independent
 - See* independent LU (ILU)
 - name, unique 483
 - secondary
 - dependent and 141
 - types of 10
- logo, VSCS 645

- logoff, terminal, (VSCS) (VM) 644
- logon
 - automatic 323
 - See also* automatic logon
 - passing request 162
 - validating request 162
- LOGON and LOGOFF, USS table, (VSCS) (VM) 632
- LOGON exits, associated LU names 157
- logon manager JCL (MVS) 654
- logon manager, defining
 - access
 - dependent logical unit 653
 - independent logical unit 653
 - APPL definition statement (operands) 656
 - configuration definition 657
 - defining the logon manager to VTAM 655
 - halting 661
 - installing 654
 - logon manager operation 653
 - logon manager, copying to system 654
 - monitoring 661
 - source statements
 - application password 659
 - buffers, maximum number 660
 - logon procedure 659
 - minimum channel link 659
 - reserved keywords 660
 - subarea configuration 657
 - subareas, maximum number 660
 - TPF applications, number 660
- logon mode table
 - APPN class of service and 558
 - class of service and 414
 - combined APPN and subarea network 520
 - entry
 - class of service 173
 - DLOGMOD operand 173
 - name 484
 - parameters 173
 - session parameters 158, 321
- logon, automatic 323
- LOGTAB operand
 - interpret table and nonswitched SDLC 251
 - interpret tables, defined 156
 - sample 251
- loop-adapter-attached devices 231
- low-entry network node
 - adjacent link station list 512
 - combined APPN and subarea network 497
 - migrating to APPN nodes 498
 - migrating to LEN nodes 497
 - overview 9
- low-entry networking node
 - See* LEN node
- LPBUF buffer pool
 - CSALIMIT start option 578
- LPBUF buffer pool (*continued*)
 - expansion 578
 - overview 584
- LU
 - addressing
 - dependent LU 141
 - dependent
 - addressing 141
 - combined subarea and APPN network 502
 - defining 141
 - overview 10
 - ownership of 141
 - secondary and 141
 - session establishment 141
 - description 139
 - independent
 - See* independent LU (ILU)
 - name, unique 483
 - secondary
 - dependent and 141
 - types of 10
- LU 6.2
 - APPCCMD macroinstruction
 - requirement for LU 6.2 196
 - APPL definition statement
 - APPC operand 196
 - AUTOSES operand 198
 - DDRAINL operand 198
 - DMINWNL operand 198
 - DMINWNR operand 198
 - DRESPL operand 198
 - DSESLIM operand 198
 - SECACPT operand 200
 - SYNCLVL operand 201
 - VERIFY operand 200
 - ATTN exit 201
 - CNOS 197
 - conversation 197
 - conversation-level verification 200
 - description 196
 - encryption facility (MVS) 168, 199
 - limited resources 201
 - LU-LU verification 199
 - mode name 197
 - session limit 197
 - session-level verification 201
 - sessions
 - automatic activation 198
 - characteristics 197
 - contention 197
 - establishment 196
 - negotiating number of 197
 - overview 196
 - sending data 197
 - terminating idle 201
 - termination 196
 - with LU 6.2 partners 197

LU 6.2 (*continued*)

- sync point services 201
- user ID verification 200

LU definition statement

- EAS operand 578
- ENCR operand (MVS) 169
- ICA connections (VM, VSE)
 - nonswitched SDLC 235
 - switched SDLC 236
- X.25 single-domain connection (VM, VSE) 244

LU names, associated 157

LU-LU verification 199

LUDRPOOL definition statement 207, 226, 292

LUGROUP operand 216

LUSEED operand 217

M

maintenance operator subsystem (MOSS) 31

major node names, list (ATCCONxx) 134

MAXBFRU operand

- coattailing and 597
- guidelines for setting 600

MAXDATA operand on PCCU definition statement 206

MAXDLUR operand on the VBUILD definition statement 504

MAXPVT operand on APPL definition statement 195

MAXSESS operand 227

MAXSUBA operand

- and extended addressing 308, 423
- extended subarea addressing 307
- gateway NCP 443
- HOSTSA start option and 307
- logon manager JCL (MVS) 654, 657
- non-extended addressing
 - communication with (SNI) (MVS, VM) 447
- non-native networks 450

message flooding prevention 351

message skeleton file 684

messages

- installing online help
 - MVS 48
 - VM 104
- module name, viewing 134
- suppression of 351
- translation of (MVS) 681
- types 351

messages, USS

- defining 352
- national language support 157, 316
- translation of (MVS) 681

migration

- subarea to APPN 497

migration data host (APPN)

- coding 499
- combined APPN and subarea network 499

migration data host (APPN) (*continued*)

- overview 8

MMS (MVS) 681

MODE operand on LINE definition statement 140

MODEENT macroinstruction

- coattailing and 597
- LANG operand (MVS) 617
- Non-SNA 3270 devices 615
- PSERVIC operand 616
- SNA 3270 devices 616
- VSCS (VM) 631

model definition 157

model major node 143, 260

model name table

- overview 157
- sample 318

MODETAB operand on APPL definition statement 173

MODIFY CDRM operator command 406

MODIFY CHKPT operator command 555

MODIFY CNOS operator command 199

MODIFY DEFINE operator command
CNOS and 199

MODIFY DR operator command 297

MODIFY ENCR operator command (MVS) 170

MODIFY NEG POLL operator command 260

MODIFY POLL operator command 260

MODIFY TABLE command 354

MODIFY TNSTAT operator command 569

module name, viewing 134

monitoring the domain

- DISPLAY command 350, 352
- messages
 - suppression of 351
 - types 351

MOSS (maintenance operator subsystem) 31

moving PUs and LUs dynamically 149, 291

MPC (multipath channel) connection for hosts

- connection for hosts 266
- tuning I/O 574

MPTN architecture 697

MSGMOD start option 134

multipath channel (MPC) connection for hosts

- connection for hosts 266
- tuning I/O 574

multiple connections for a type 2.1 node 224

multiple operator consoles (MVS) 353

multiple-network environment

- activating resources 493
- combined APPN and subarea network 513
- disjointed networks 513
- network management with NetView 493
- terminating sessions 493

multiple-network, verification 494

Multiprotocol Transport Networking (MPTN) 697

MVS data sets 21

MVS Message Service 681
 MVS, defining VTAM to
 adding channel-attached devices 32
 DATAMGT macroinstruction 32
 DATASET macroinstruction 31
 IODEVICE macroinstruction 32
 routing code, changes 32
 SECONSLE macroinstruction 32
 MVS, installing VTAM under
 data sets for installation
 configuration restart data set 27
 NODELST data set 29
 O/S data sets used by VTAM 21
 SYS1.VTAMLIB (required) 25
 SYS1.VTAMLST (required) 24
 VTAM- and NCP-specific data sets 24
 defining VTAM to MVS
 channel-attached devices, adding 32
 DATAMGT macroinstruction 32
 IODEVICE macroinstruction 32
 name system data sets 31
 routing codes changes 32
 SECONSLE macroinstruction 32
 installing, using distribution tape 34
 NCP-specific data sets
 NCP dump data set 30
 NCP load library 29
 start procedure VTAM, defining
 name of start procedure 34
 sample of start procedure JCL 34
 start procedure name 34
 SYS1.PROCLIB 34
 MVS/ESA operator functions for TSO 624

N

name
 class-of-service 484
 logon mode 484
 subarea-unique, single-domain 136
 name assumption 460
 name restrictions, single-domain 136
 national language support
 defining TSO to (MVS) 623
 description 157, 316
 LANG operand (MVS) 316, 617
 LANGTAB operand 316
 MVS message service (MVS) 681
 SYS1.SISTDAT2 data set (MVS) 22
 TSO/VTAM end-user messages (MVS) 623
 native operand for subnetwork boundaries 549
 NCP
 activating 371
 amount of data sent to VTAM
 MAXBFRU operand 206
 APPN connections
 assigning addresses 535

NCP (*continued*)
 APPN connections (*continued*)
 exhausting session control blocks 535
 maximum number of sessions 535
 attaching to VTAM
 overview 138, 202
 automatic logon (SNA sessions) 158, 323
 backup of owner 432
 BUILD definition statement 138
 channel unit address 208
 channel-attached to VTAM 207
 CMC configuration 145
 coattailing for 597
 congestion, reducing 594
 connecting hosts through 267
 connection networks 532
 controlling in CMC environment
 channel-attachment major node 285
 dynamic switched major node 144
 NCP major node 285
 resource ownership 286
 switched major node 286
 CWall threshold condition 569
 deactivating 381
 defining
 BUILD definition statement 204
 GROUP definition statement 209
 HOST definition statement 206
 PCCU definition statement 202
 dial-in (to VTAM)
 overview 216
 V.25 bis protocol (VM, VSE) 216
 dial-out (to NCP)
 automatic 215
 manual 216
 dump
 automatic (MVS) 203
 dynamic 382
 performance, improving 133
 remote 214
 selecting a dump disposition 385
 selecting a dump file 385
 static (destructive) 382
 storing 204
 dynamic control facilities 207
 enhanced remote controller support 214
 enterprise system connection channel 230
 generation
 overview 362
 increasing data transfer speeds 594
 independent LU 226
 line speed, reducing buffer use 595
 link-attached NCP
 activating 371
 LINE definition statement 212
 link and link-station definition 213
 overview 212

- NCP (*continued*)
 - link-attached NCP (*continued*)
 - PU definition statement 212
 - XCA (3172) and 212
 - links
 - activating 429
 - deactivating 429
 - load library 29
 - load module
 - verifying 204
 - loading
 - overview 362
 - performance, improving 133
 - reloading, automatic (scheduled) 366
 - remote 214
 - loading and activating using switched link
 - PCCU definition statement 377
 - VARY ACT operator command 377
 - monitoring 379
 - NCP definition statement 202
 - ownership of 434
 - owning discontinuous domains 432
 - pacing 159
 - overview 330
 - reducing congestion 595
 - parallel transmission groups 209
 - peripheral nodes, attached to 250
 - PIU size
 - MAXDATA, setting for 206
 - polling
 - reduction due to slowdown 595
 - recording and sending line errors to VTAM 379
 - reload
 - automatic 203
 - remote 214
 - sharing in combined APPN and subarea
 - network 508
 - sharing of 434
 - SLODN tuning statistic 569
 - slowdown, controlling 594
 - storage, displaying 380
 - switched SDLC 215
 - token ring connections (NTRI) 256
 - transmission groups, VTAM and NCP 209
 - tuning I/O for 570
 - tuning statistics 569
 - V.25 bis protocol modem (VM, VSE) 216
 - X.25 lines 143
- NCP Packet Switching Interface (NPSI)
 - single-domain connections 143
- NCP, peripheral nodes attached to
 - binary synchronous (BSC) connection 259
 - nonswitched SDLC connection 250
 - switched SDLC connection 251
 - token-ring connection 256
 - X.21 connection (VSE) 143, 259
- NCPBUFSZ start option 133
- NETID
 - ISO standards for 128
 - registering 128
 - start option
 - multiple-network 441
 - single-domain 128
- NETSRVR definition statement
 - SLUINIT operand 544
- NETTYPE operand on PORT definition statement 247
- NetView program
 - alias name translation
 - adjacent SSCP tables and 473
 - defining 443
 - overview 483
 - APPL definition statement 442
 - defining to VTAM 442
 - network management with 493
 - XCA (3172) and 231
- network accessible units 10
- Network Control Program
 - See* NCP
- network ID, dynamic 451
- network identification 443
- network node server (APPN)
 - list
 - defining 544
 - replacing 545
 - sample 545
 - overview 7
 - resource registration 554
 - supporting SLU-init sessions 544
- network node, composite
 - See* composite network node
- network nodes (APPN)
 - overview 5
- network qualified names
 - CDRSCs and 458
 - concurrent sessions with independent LUs 219
 - defining the NetView program and 442
 - overview 474
- Network Registry, SNA 128
- network routing 151
- network topology database (APPN)
 - See* topology database (APPN), network
- NMVTLOG start option 133
- NODELST data set 342
- nodes
 - APPN 4
 - subarea 3
 - subarea and APPN 8
 - types of 3
- NODETYPE start option 498, 499, 500, 525
- non-native network connection 250, 490
- non-SNA devices
 - cluster controllers
 - defining 228

non-SNA devices (*continued*)
 data-stream characteristics 196
 deactivating 228
 defining 228
 loop-adapter-attached 231
 nonswitched SDLC connection
 ICA connected (VM, VSE) 234
 multiple domain
 ES/9000 series to domain (VM, VSE) 278
 NCP to domain 267
 single domain
 NCP to peripheral node 250
 VTAM to peripheral node 234
 NPSI (NCP Packet Switching Interface)
 single-domain connections 143
 NQNM MODE start option 395, 458
 NSRU (unsolicited network services request unit) 359
 NTRI (NCP/Token-Ring Interconnection) 256
 defining a connection to a connection network 532
 NUMILU operand on the LUDRPOOL definition statement 226

O

Online Message Facility (MVS, VM)
 installing
 MVS 48
 VM 104
 using 352
 operands, dynamically changing 149, 292
 operator commands
 DISPLAY BFRUSE 567
 MODIFY NEG POLL 260
 MODIFY POLL 260
 MODIFY TNSTAT 569
 OPNDST macroinstruction
 OPTCD operand 163
 OPTCD operand on OPNDST macroinstruction 163
 outages, reducing application program 174

P

pacing control for sessions
 application program 337
 congestion, reducing 595
 congestion, route 311
 defining values
 overriding defined pacing counts 333
 primary-secondary 333
 secondary-primary 332
 one-stage pacing 331
 overview 159
 pacing counts, overriding 333
 setting values 159, 339
 two-stage pacing 331

pacing of data flow 159
 pacing, adaptive session 334
 pacing, virtual route 310
 packet switched data network (PSDN) (VM, VSE)
 DTE-to-DTE single-domain connection 247
 ICA connected 243
 optional user facilities 246
 sample 245
 VTAM Common Network Services (VCNS)
 (VM) 246
 page frames (VM) 649
 parallel sessions 163, 481
 parallel transmission groups (TG)
 NCP, channel-attached 209
 PARSESS operand on APPL definition statement 163, 481
 partitions for VTAM (VSE)
 sizes 115
 SUPVR macroinstruction 114
 password verification, two-way 271
 path definition
 dynamic update
 description and sample 313
 restrictions 314
 PATH definition statement
 dependent LU requester, defining 504
 DIALNO operand
 NTRI connection 258
 V.25 bis (VM, VSE) 216
 DLCADDR operand
 NTRI connection 258
 GID operand 254
 NTRI connection, for 258
 PID operand 254
 VTAM switched SDLC connection 254
 X.21 direct call (VSE) 238
 path update, dynamic 313
 paths
 forcing a route (APPN) 685
 logical 153
 multiple-domain environment 421
 multiple-network environment 475
 network routing for subarea nodes 151
 physical 151
 routing in a subarea network 414
 single-domain environment 151
 PAUSE operand on GROUP definition statement 260
 PCCU definition statement
 AUTOSYN operand 204
 description 202
 multiple 203
 sample 202
 sample, channel-attached 208
 SUBAREA operand 203
 VFYC operand 204
 VFYLM operand 204

- PCF (Programmed Cryptographic Facility) (MVS) 168
- peak traffic, unusual 159
- performance
 - APPN
 - checkpointing 555
 - improving search performance 553
 - collecting data 569
 - introduction 565
 - NCP slowdown 567, 594
 - NCP tuning statistics 569
 - VSCS, controlling (VM) 636
- performance, system
 - APPN
 - improving search performance 553
 - buffer pools specifications 583
 - common service area (CSA) 578
 - definition 565
 - input/output (I/O) operations 570
 - session pacing 339
- peripheral link 152
- peripheral node, type 2.1
 - See* PU
- peripheral nodes
 - attaching to NCP 142
 - attaching to VTAM 142, 228
 - controlling 145
 - loop-adapter-attached devices 231
 - SNA major node, local 229
- peripheral subnetwork boundaries 548
- PERSIST operand on ACB macroinstruction (MVS) 193
- persistent LU-LU sessions (MVS)
 - description 193
 - sync point and 201
- physical unit
 - adding dynamically 291
 - attaching to VTAM 142, 228
 - BSBUF buffer pool start option 219
 - deleting dynamically 291
 - loop-adapter-attached 231
 - moving dynamically 291
 - non-SNA devices 228
 - SNA devices
 - defining 229
 - operator-assisted connect 230
- type 2.1
 - casual connection 408
 - channel, defining 140
 - characteristics 139
 - combined APPN and subarea network 497
 - CPNAME operand and 139
 - identifying 139
 - independent LU 218
 - local area network 140
 - MODE operand and 140
 - NCP-attached 249, 259
 - non-native network connection 250
- physical unit (*continued*)
 - type 2.1 (*continued*)
 - non-SNA major node 228
 - restrictions, alias name 219
 - restrictions, independent LU 219
 - session capabilities 218
 - session control block, maintaining 219
 - session establishment 139
 - sessions, maximum number 227
 - XID services 140
 - types of 10
- PHYSRSC operand on PU definition statement
 - OWNER operand and 430
 - single-domain, in a 375
- PID operand on PATH definition statement 254
- POLIMIT operand on GROUP definition statement 260
- polling terminals (NCP)
 - changing parameters for 260
 - reduction due to slowdown 595
- PORT definition statement
 - NETTYPE operand 247
- predefined cross-domain resource 463
- printer (VSCS) (VM)
 - logged on to VSCS 642
 - TIMEREL (VSCS option) 642
 - VSCS printer table 642
 - warning, printer problem 643
- printer problem, warning, VSCS (VM) 643
- private storage, allocating 195
- problem analysis
 - IPCS CLISTS 22, 28
- problem diagnosis
 - alerts to NPDA 133
 - module name, viewing 134
 - NMVTLOG start option 133
 - SDLC statistical MDRs 134
 - start options 133
 - time interval notification 134
 - trace facility 134
- program operators
 - associating with USS tables 353
 - defined 358
 - making decisions on setting up 358
- Programmed Cryptographic Facility (PCF) (MVS) 168
- protection of subarea dial 271
- PSDN (packet switched data network) (VM, VSE)
 - DTE-to-DTE single-domain connection 247
 - ICA connected 243
 - optional user facilities 246
 - sample 245
 - VTAM Common Network Services (VCNS) (VM) 246
- PSERVIC operand 616
- PU
 - adding dynamically 291
 - attaching to VTAM 142, 228

PU (continued)

- BSBUF buffer pool start option 219
- deleting dynamically 291
- loop-adapter-attached 231
- moving dynamically 291
- non-SNA devices 228
- SNA devices
 - defining 229
 - operator-assisted connect 230
- type 2.1
 - casual connection 408
 - channel, defining 140
 - characteristics 139
 - combined APPN and subarea network 497
 - CPNAME operand and 139
 - identifying 139
 - independent LU 218
 - local area network 140
 - MODE operand and 140
 - NCP-attached 249, 259
 - non-native network connection 250
 - non-SNA major node 228
 - restrictions, alias name 219
 - restrictions, independent LU 219
 - session capabilities 218
 - session control block, maintaining 219
 - session establishment 139
 - sessions, maximum number 227
 - XID services 140
- types of 10
- PU definition statement
 - ANS operand 252
 - CONNTYPE operand 534
 - CPCP operand 533
 - CPNAME operand
 - type 2.1 PU and 139
 - ICA connections (VM, VSE)
 - nonswitched SDLC 234
 - switched SDLC 236
 - IDBLK operand 139
 - IDNUM operand 139
 - PHYSRSC operand
 - OWNER operand and 430
 - single-domain, in a 375
 - X.25 single-domain connection 244
- PU operand on GROUP definition statement 259
- PUDRPOOL definition statement
 - NCP type 2.1 and 207, 249, 292
- purging a communication controller disk 370

R

- RDEVBLK (VM control blocks) (VM) 650
- reactivation, forced 356
- reconfiguration, dynamic
 - effect on WARM start 347

reconfiguration, dynamic (continued)

- frame relay lines, PUs on 298
- implicit dynamic reconfiguration 300
- overview 149, 291
- samples 301
- VARY ACT, UPDATE technique 292
- VARY DRDS technique 295
- recovery (MVS)
 - cryptography and 175
 - description 174
- recovery operations
 - link-attached NCP, for 212
- registering resources 554
- reloading a controller, automatic 366
- renaming NCP load modules 368
- resource control, cross-domain
 - CDRSC definition statement 398
 - controlled by another VTAM domain 397
 - cross-domain resource major node 397
 - dynamic adjacent SSCP table 400
 - locating cross-domain resources 398
 - network resources for XRF 398
 - USERVAR with XRF 398
- resource control, single-domain
 - activating 348
 - automatic activation of cross-subarea links 371
 - order of activation 349
 - deactivating
 - automatic deactivation of cross-subarea links 355
 - forced deactivation 356
 - forced reactivation 356
 - immediate deactivation 356
 - normal deactivation 355
 - order of deactivation 355
- resource definition statements
 - coding 136
 - sample 136
 - storing 137
 - updating 137
- resource definition, dynamic (model name) 157
- resource ownership, SSCP 432
- resource registration 554
- resource resolution table (RRT)
 - description 362
 - time stamps 204
- resource, restoring (MVS, VSE)
 - defined 342
 - NODELST data set 342
 - WARM start 343
- resources in other networks
 - adjacent SSCP (ADJSSCP) table, defining 467
 - adjacent SSCP lists for CDRSCs 400
 - CDRM ownership
 - changing by operator 406
 - changing dynamically 406
 - verifying 406

- resources in other networks (*continued*)
 - combined APPN and subarea network 508
 - defining 395, 508
 - definition statement
 - VFYOWNER operand 406
 - description 467
 - dynamic 465
 - locating
 - adjacent SSCP table 398
 - ADJCDRM definition statement 401
 - CDRSC routed to each SSCP 399
 - cross-network 467
 - multiple-network configuration 468
 - multiple-network environment 458
 - performance improving 404
 - predefined 462
 - SSCPDYN start option 404
 - SSCPORD start option 404
- response time
 - channel-to-channel connection 604
 - coattailing and 577, 595
 - DELAY operand and 596, 599
 - overview 565
- RESSCB operand on the LU definition statement 227
- restart a host 433
- restoring resource (MVS, VSE)
 - defined 342
 - NODELST data set 342
 - WARM start 343
- restricted materials 695
- restrictions
 - active VSCS (VM) 628
 - addressing 307
 - independent LU 219
 - name, single-domain 136
 - subarea addressing 426
- RESUSAGE start option 559
- route
 - assigning session traffic to a 414
 - explicit 151
 - extension 154
 - planning 418
 - virtual 153
- route congestion 311
- routing
 - APPN
 - broadcast search 552
 - class of service and 558
 - directed search 551
 - directory services 550
 - forcing a route 685
 - improving efficiency 553
 - local topology database 16
 - network topology database 16
 - overview 16
 - route selection 16, 558
 - topology database 550
 - routing (*continued*)
 - code changes (MVS) 32
 - NETDA/2 (route planning tool) 419
 - network 151
 - overview 15
 - route calculation (APPN)
 - overview 18
 - route selection (APPN)
 - overview 18, 558
 - routing tables 152
 - selection of route
 - APPN 16
 - subarea 15
 - topology database (APPN)
 - overview 16
 - routing subarea node
 - address incompatibility 423
 - addressing methods 422
 - extended addressing compatibility 425
 - extended network addressing 423
 - path definition, subarea 421
 - subarea addressing restrictions 426
 - routing, cross-network
 - communication between network 474
 - COS tables 479
 - extended subarea address gateway 474
 - extended subarea addressing 474
 - gateway NCP 474
 - gateway VTAM (MVS, VM) 474
 - virtual route, default 479
 - routing, network, subarea nodes
 - addressing compatibility
 - ERLIMIT operand 309
 - extended subarea 309
 - network configuration 309
 - SALIMIT operand 309
 - addressing restrictions 307
 - compatibility, non-extended address 307
 - element address 306
- RRT (resource resolution table)
 - description 362
 - time stamps 204

S

SALIMIT operand 309

samples

- channel-to-channel
 - tuning statistics report 572
- display buffer use command output 567
- multipath channel
 - tuning statistics report 574
- networks
 - multiple-domain 389
 - multiple-network 438
 - single-domain 126

- samples (*continued*)
 - SNA controller
 - tuning statistics report 570
 - SAW (session awareness) data, filtering 360
 - scheduled NCP reload 366
 - SCIP (session control in-bound processing exit) 638
 - SCIPNUM (VSCS) (VM) 638
 - SCRIPTIM (VSCS) (VM) 638
 - SDLC (synchronous data link control)
 - nonswitched connection
 - ICA-attached (VM, VSE) 234
 - NCP to peripheral nodes 250
 - switched connection
 - dial-in (to VTAM) 236, 253
 - dial-out (from VTAM) 254
 - ICA to peripheral node (VM, VSE) 235
 - NCP to peripheral node 251
 - testing links 380
 - SDLC statistical MDRs 134
 - SDLCMDRS start option 134
 - search reduction 131
 - searches
 - broadcast 552
 - controlling
 - combined APPN and subarea network 512
 - order of 512
 - subarea network searches 513
 - directed 551
 - eliminating, for unavailable resources 131
 - improving performance 555
 - limiting broadcast searches 513
 - limiting SNI searches 513
 - limiting subarea searches 513
 - performance enhancements 553
 - reducing, for unavailable resources 131
 - time interval, controlling 392
 - types of 551
 - SECACPT operand on APPL definition statement 200
 - security (VSCS) (VM) 633
 - security facilities
 - confidential data 173
 - single-domain 168
 - security of data
 - application program 173
 - conversation-level 199
 - session-level 199
 - security of data (MVS)
 - CDRM key 665
 - cross-domain cryptography 665
 - filing secondary logic unit (SLU) keys 663
 - single-domain encryption 663
 - security verification 271
 - selective termination of idle LU 6.2 sessions
 - description 201
 - non-LU 6.2 sessions and 201
 - service (VM)
 - overview 86
 - session awareness (SAW) data, filtering 360
 - session control in-bound processing exit (SCIP) 638
 - session initiation request 459
 - session management exit routine (SME)
 - adjacent SSCP selection 427
 - alias selection function 482
 - for multiple-domain sessions 427
 - session accounting 330
 - session authorization 329
 - when called 329
 - session pacing
 - See* pacing control for sessions
 - session-partner resource 157
 - sessions
 - accounting for cross-domain 330
 - acquiring 163
 - adjacent SSCP selection 404
 - application program, controlling 323
 - APPN
 - setting maximum number 535
 - associated LU table 157
 - authorize, establish, or terminate 155
 - authorizing cross-domain 329
 - automatic logon 323
 - backup
 - extended recovery facility (XRF) 174
 - between logical units 155
 - changing an LU's automatic logon specification 325
 - combined APPN and subarea network 512
 - control blocks
 - controlling use of (APPN) 535
 - exhausting (APPN) 535
 - controlling 427
 - CP-CP (APPN)
 - overview 12
 - cross-network session control
 - gateway SSCP 445
 - data flow control 159
 - disruption, BSC 434
 - establishing 427
 - establishing and terminating with commands 354
 - establishment
 - dependent LU 141
 - type 2.1 PU 139
 - gateway path selection 475
 - gateway session accounting 492
 - LOGON exits 157
 - logon mode table 158
 - logon, automatic 323
 - LU-LU
 - overview 13
 - model name table 157
 - number of half-sessions 492
 - overview 11

- sessions (*continued*)
 - pacing of data flow 159
 - parallel 163
 - persistent LU-LU (MVS) 193
 - protocol 158
 - security
 - LU 6.2 session-level LU-LU verification 199
 - session accounting exit routine 329
 - session authorization exit routine 329
 - session management exit routine 427
 - session parameters, devices 631
 - SLU-init in network node server (APPN) 544
 - SNA network interconnection 492
 - SSCP-SSCP 11
 - tailoring
 - authorize or establish 155
 - between logical units 155
 - routines used 155
 - tables used 155
 - terminating in a multiple-network environment 493
 - termination 354, 493
 - USERVAR, session failure 181
- SET QUICK DSP 650
- SET REMOTE (VM) 650
- SET RESERVE (VM) 649
- SET SHARE RELATIVE 1000 651
- SFBUF buffer pool
 - guidelines for 589
 - overview 584
- SGA (system GETVIS area) (VSE)
 - limiting VTAM use of 579
 - monitoring 567
- SGALIMIT start option (VSE) 579
- shadow resource
 - application programs 125
 - CDRSCs 397, 398
 - overview 407
 - takeover and 433
- SHOLD operand on GROUP definition statement 237
- shut down data transmission 311
- sift-down coding 137
- single-domain name restrictions 136
- single-domain network
 - overview 125, 389
 - sample 126
- skeleton file, message 684
- SLODN tuning statistic 569
- slowdown condition
 - analyzing 569
 - controlling NCP 594
- SLUINIT operand on NETSRVR definition statement 544
- SME (session management exit routine)
 - adjacent SSCP selection 427
 - alias selection function 482
 - for multiple-domain sessions 427
- SME (session management exit routine) (*continued*)
 - session accounting 330
 - session authorization 329
 - when called 329
- SNA controllers
 - See cluster controller or communication controller
- SNA devices
 - cluster controllers
 - defining statically 229
 - loop-adapter-attached 231
- SNA network interconnection (SNI) (MVS, VM) 440
 - See also multiple-network environment
- SNA Network Registry 128
- SNASVCMG default class of service 557
- SNI (SNA network interconnection) (MVS, VM) 440
 - See also multiple-network environment
- SNVC start option 549
- SONLIM start option 591
- SORDER start option 501, 512
- SPBUF buffer pool
 - overview 584
- speed of lines
 - See lines, speed
- SRCHRED start option 132
- SRCLEAR operand on the MODIFY RESOURCE
 - command 132
- SRCOUNT start option 132
- SRTIMER start option 132
- SSCP
 - identifier 127
 - overview 4
 - owning 484
- SSCP resource ownership 432
- SSCP takeover
 - combined APPN and subarea network 509
 - description 432
- SSCP, adjacent
 - See adjacent SSCP
- SSCP, gateway (MVS, VM)
 - CDRM definition statement 456
 - defining 445
 - GWPATH definition statement 456
 - NETWORK definition statement 455
- SSCPDYN start option 404
- SSCPID start option 127, 441
- SSCPNAME start option 128, 441
- SSCPORD start option 404
- SSEARCH start option 502, 513
- start option list (ATCSTR00) 127, 129, 139
- start option list, backup (LISTBKUP) 129, 130
- start options
 - APPN
 - APPNCOS 549, 558
 - BN 549
 - BNDYN 549
 - BNORD 549
 - CDSERVR 554

start options (*continued*)APPN (*continued*)

- CONNTYPE 497
- CPCP 525
- DIRSIZE 555
- DIRTIME 555
- DYNADJCP 526
- INITDB 556
- NODETYPE 498, 499, 525
- RESUSAGE 559
- SNVC 549
- VERIFYCP 200
- VRTG 517
- VRTGCPCP 517

AUTHLEN 227

AUTORTRY 326

AUTOTI 326

CMPMIPS 164

combined APPN and subarea network 501

DISCNTIM 256

DYNLU 526

ENHADDR 306

gateway VTAM 441

IOINT start option 134

IOPURGE 392

MSGMOD start option 134

- alerts to NPDA 133
- VTAM trace facility 134

multiple-domain

- performance improvement 391
- performance problems 391

multiple-network

- GWSSCP 442
- HOSTPU 441
- NETID 441
- SSCPID 441
- SSCPNAME 441

NODETYPE 498, 499, 500, 525

NQNMODE 395

overriding 130

overview 125, 127

performance 391

prompting for start options 129

routing and 559

single-domain

- creating start options 130
- GCS ACCESS command (VM) 131
- host physical unit name 128
- LISTBKUP 129, 130
- NCPBUFSZ 133
- NETID 128
- sample 130
- source of start options 129

SORDER start option 501, 512

SRCHRED start option 131

SRCOUNT start option 131

start options (*continued*)

- SRTIMER start option 131
- SSCP identifier 127
- SSEARCH start option 502, 513
- SWNORDER 139, 253
- VSCS, for (VM) 633
- XNETALS start option 250

start procedure

- name consistency (MVS) 34
- sample JCL (MVS) 34
- sample JCL (VSE) 117
- SYS1.PROCLIB (MVS) 34

start-stop devices, communicating with 196

start, warm (MVS, VSE)

- configuration restart 342
- initial status and 137

starting

- application program 172
 - name of application and 162
- VSCS, sample (VM) 629

VTAM

- configuration list 134
- creating start options 130
- general start procedures, MVS 341
- general start procedures, VM 342
- general start procedures, VSE 342
- problem diagnosis 133
- prompting for start options 129
- sample start options, single-domain 130
- source of start options 129
- start options 127

status information, resource (MVS, VSE)

- defined 342
- NODELST data set 342
- WARM start 343

storage

- allocating private for application program 195
- balancing storage needs 565
- calculating waste 568
- displaying NCP storage 380
- dynamic storage (VM)
- monitoring buffer use 567
- shortage of 579
- storage management control, VSCS (VM)
 - STCHKTM option 633
 - STRELTm option 633
- virtual storage requirements
 - MVS 21
 - VM 51
 - VSE 107
- VTAM uses 565

storing resource definition statements 137

subarea node address

- address incompatibility 423
- addressing methods 422
- extended addressing compatibility 425

- subarea node address (*continued*)
 - extended network addressing 423
 - path definition, subarea 421
 - subarea addressing restrictions 426
- subarea nodes, network routing for
 - dynamic path update 313
 - extended network and subarea nodes 309
 - nodes, non-extended and extended 307
 - PATH definition statements 151
- SUBAREA operand on PCCU definition statement 203
- subarea to network data control 310
- subarea-unique name, single-domain 136
- substituting COS parameters 416
- SUBSTUT operand on COS macroinstruction 416
- SUPVR macroinstruction (VSE) 114
- switched SDLC connection
 - dial-in to VTAM 236, 253
 - dial-out from VTAM 254
 - dynamic definition of 143, 260
 - ICA (VM, VSE) 235
 - multiple-domain
 - ES/9000 series to domain (VM, VSE) 279
 - NCP to domain 269
 - PATH definitions
 - VTAM connections 254
 - single-domain
 - NCP to peripheral node 251
 - VTAM to NCP 215
 - VTAM to peripheral node 235
- SWNORDER start option 139, 253
- sync point services, LU 6.2
 - description 201
 - persistent LU-LU sessions and (MVS) 201
- synchronous data link control (SDLC)
 - nonswitched connection
 - ICA-attached (VM, VSE) 234
 - NCP to peripheral nodes 250
 - switched connection
 - dial-in (to VTAM) 236, 253
 - dial-out (from VTAM) 254
 - ICA to peripheral node (VM, VSE) 235
 - NCP to peripheral node 251
 - testing links 380
- SYNCLVL operand on APPL definition statement 201
- SYS1.DSDB1 data set 21, 26
- SYS1.DSDB2 data set 22, 26
- SYS1.DSDBCTRL data set 22, 26
- SYS1.SISTCLI0 data set 22
- SYS1.SISTCLS1 data set 22, 28
- SYS1.SISTDAT1 data set 22
- SYS1.SISTDAT2 data set 22, 684
- SYS1.SISTDEBUG data set 22
- SYS1.SISTMSG0 data set 23
- SYS1.SISTPNL0 data set 23
- SYS1.SISTPNL1 data set 23

- SYS1.SSPLIB data set 23, 30
- SYS1.TRACE data set 23
- SYS1.TRSDDB data set 23, 26
- SYS1.VTAMLIB data set 25
- SYS1.VTAMLST data set 24
- SYS1.VTAMLST, start options 130
- system GETVIS area (SGA) (VSE)
 - limiting VTAM use of 579
 - monitoring 567

T

- table, replacing 354
 - See also* dynamic table replacement
- tailoring VSCS 631
- takeover of resources
 - combined APPN and subarea network 509
 - description 432
- TCAM with TSO (MVS) 624
- TCAS (terminal control address space)
 - See* terminals
- TCP/IP
 - See* LAN (local area network)
- TERMINAL definition statement
 - BSC single-domain connection 248
- terminal logoff (VSCS) (VM) 644
- terminal logon (VSCS) (VM) 643
- terminals
 - 3277 228
 - 8775 231
 - control address space (TCAS) 612
 - non-SNA 228
 - polling
 - changing parameters for 260
 - taking off-line 228
- TIC (token-ring interface connector)
 - address 257
 - dump 386
- time sharing
 - buffers 620
 - cataloged procedures 620
- time stamps
 - RRT and load module 204
- timed IPL 366
- TNSTAT start option 569
- token-ring connection
 - APPN
 - sample 3172 interconnect controller 539
 - multiple subsystems (VM, VSE) 242
 - NCP/Token-Ring Interconnection (NTRI) 256, 273
 - VTAM, multiple domain
 - ICA connections (VM, VSE) 278, 283
 - XCA connections 274
 - VTAM, single domain
 - ICA connections (VM, VSE) 239
 - XCA connections 231

- token-ring interface connector (TIC)
 - address 257
 - dump 386
- topology and routing services (TRS) 550
- topology database (APPN)
 - checkpointing 555
 - creating 550
 - improving performance 555
 - local
 - overview 16
 - network
 - overview 16
 - overview 16
 - topology exchange 550
- TP (transmission priority) 154
 - authorized, for LEN connections 227
- TPF (Transaction Processing Facility) (MVS)
 - access
 - dependent logical unit 653
 - independent logical unit 653
 - APPL definition statement (operands) 656
 - configuration definition 657
 - defining the logon manager to VTAM 655
 - halting 661
 - installing 654
 - logon manager operation 653
 - logon manager, copying to system 654
 - monitoring 661
 - source statements
 - application password 659
 - buffers, maximum number 660
 - logon procedure 659
 - minimum channel link 659
 - reserved keywords 660
 - subarea configuration 657
 - subareas, maximum number 660
 - TPF applications, number 660
- trace facility 134
- trace formatter panels, verifying 43, 101
- tracing
 - VSE files for 113
- trademarks xxvii
- Transaction Processing Facility (TPF) (MVS)
 - access
 - dependent logical unit 653
 - independent logical unit 653
 - APPL definition statement (operands) 656
 - configuration definition 657
 - defining the logon manager to VTAM 655
 - halting 661
 - installing 654
 - logon manager operation 653
 - logon manager, copying to system 654
 - monitoring 661
 - source statements
 - application password 659
 - buffers, maximum number 660
- Transaction Processing Facility (TPF) (MVS) (*continued*)
 - source statements (*continued*)
 - logon procedure 659
 - minimum channel link 659
 - reserved keywords 660
 - subarea configuration 657
 - subareas, maximum number 660
 - TPF applications, number 660
- translation, alias name
 - adjacent SSCP tables and 473
 - alias selection function (VTAM) 482
 - defining 485
 - LU 6.2 488
 - NetView 442
 - overview 483
- transmission groups (TG)
 - APPN
 - installing the default profiles 25
 - boundary function-based (APPN) 527
 - IBMTGPS file (APPN) 25
 - parallel
 - NCP, channel-attached 209
 - overview 528
 - parallel sessions 312
 - PCCU definition statement 210
 - PU definition statement 209
 - VR-based 515
- transmission priority (TP) 154
 - authorized, for LEN connections 227
- TRL major node 536
- TRLE operand on the PU definition statement 536
- TRS (topology and routing services) 550
- TSO Operator (MVS) 624
- TSO/VTAM (MVS)
 - 2741, TWX, or WTTY 619
 - 3270 characteristics 615
 - 3270 large screen considerations 622
 - 3790/3270 configuration 618
 - definition 611
 - estimating active sessions for (EAS) 578
 - exit routines 621
 - implementing 621
 - interpret table, defining 620
 - LFBUF buffer pool and 589
 - messages, end-user 623
 - multiple-domain network 613
 - performance 622
 - security messages 621
 - session parameters 614
 - single-domain network 612
 - TCAS application, defining 612
 - TCAS program properties, defining 620
 - translation of messages 681
 - translation table 621

TSO/VTAM versus TCAM with TSO (MVS) 624

tuning I/O operations

- analyzing tuning statistics 570
- information gathering 570

tuning statistics

- blocking outbound PIUs, analyzing 577
- channel-to-channel adapter 571
- data transfer, analyzing 576
- gathering 569
- host processor use, analyzing 576
- I/O buffer size, analyzing 576
- read attentions (RDATN) 576
- record format for SNA controller 570
- slowdown, analyzing 569
- SNA controllers 570

tuning statistics analysis

- blocking outbound PIUs, analyzing 577
- channel-to-channel adapter 571
- data transfer, analyzing 576
- gathering 569
- host processor use, analyzing 576
- I/O buffer size, analyzing 576
- read attentions (RDATN) 576
- record format for SNA controller 570
- slowdown, analyzing 569
- SNA controllers 570

tuning your network

- APPN
 - improving search performance 553
- buffer pools specifications 583
- common service area (CSA) 578
- definition 565
- input/output (I/O) operations 570
- session pacing 339

type 1 physical units 149

See also PU

type 2 physical units 149

See also PU

type 2.1 peripheral node, attaching

See PU

type 2.1 physical unit

See PU

type 4 subarea node 145

type 5 subarea node 145

UNITSZ operand on HOST definition statement 597

U

unsolicited network services request unit (NSRU) 359

UPARM 559, 685

updating resource definition statements 137

user ID verification, LU 6.2 200

user variable (USERVAR), classes

- class
 - automatic 177
 - user-managed 176

user variable (USERVAR), classes (*continued*)

- type
 - dynamic 178
 - static 177
 - volatile 178

USERVAR (user variable), classes

- class
 - automatic 177
 - user-managed 176
- type
 - dynamic 178
 - static 177
 - volatile 178

USS commands

- defined 352
- tables 632

USS messages

- defining 352
- national language support 157, 316
- translation of (MVS) 681

USS table

- associating with program operators 353
- changing, modifying 352, 354
- logon and logoff requests from logical units 156
- national language support 157, 316
- operation-level 352
- session-level 156

USS table, LOGON and LOGOFF, (VSCS) (VM) 632

V

V.25 bis protocol modem (VM, VSE) 216

VCNS (MVS, VM)

- overview 195
- X.25 PSDN and (VM) 246
- XCA (3172) and 195

VCNS operand on APPL definition statement (MVS, VM) 195

VCPARMS definition statement (VM, VSE)

- X.25 single-domain connection 244

VERALSID operand on PU definition statement 253

verification, LU-LU 199

verification, user ID, LU 6.2 200

VERIFYCP start option 200

verifying a multiple-domain network 435

verifying single-domain network 160

VFYC operand on PCCU definition statement 204

VFYLM operand on PCCU definition statement 204

VFYOWNER operand on CDRSC definition statement 406

virtual node 530

virtual route (VR)

- coattailing and 604
- description of 153

virtual route pacing 310

- virtual storage, data space 115
- VIT (VTAM internal trace) analysis tool
 - data sets 22
 - installing 35, 96
- VM
 - CP system name table (DMKSNT) 52
 - directory entries, creating 61
 - directory options
 - DIAG98 option 649
 - SET REMOTE 650
 - SET RESERVE 649
 - installing VTAM under 51
 - servicing VTAM 86
 - SET QUICK DSP 650
 - SET SHARE RELATIVE 1000 651
 - storage layout of VTAM 52
- VM control blocks (RDEVBLK) (VM) 650
- VMSES/E 51
- VN=YES operand on the ADJCP definition
 - statement 533
- VNGROUP operand on the LINE definition
 - statement 532
- VNGROUP operand on the PORT definition
 - statement 533
- VNNAME operand on the LINE definition
 - statement 532
- VNNAME operand on the PORT definition
 - statement 533
- VR (virtual route)
 - coattailing and 604
 - description of 153
- VR-based TG 515
- VRTG operand on the CDRM definition statement 517
- VRTG start option 517
- VRTGCPCP operand on the CDRM definition
 - statement 517
- VRTGCPCP start option 517
- VSAM configuration restart file (MVS, VSE)
 - associating with major nodes 347
 - CONFGDS operand 347
 - CONFGPW operand 347
 - MVS installation 27
 - VSE installation 111
- VSCS (VM/SNA console support)
 - accounting information 647
 - CHANGE command
 - start options and 630
 - defining VSCS as a VTAM application 627
 - directory entry of VSCS 628
 - DTIGEN macroinstruction 633
 - dumps, VSCS, controlling 635
 - multiple VSCS applications 627
 - performance, controlling 636
 - SCIP recovery 638
 - storage management, controlling 633
 - preparing for GCS 629
- VSCS (VM/SNA console support) (*continued*)
 - printers used by VSCS 642
 - start options (VSCS) 639
 - tailoring VSCS 631
 - terminal logoff (VSCS) 644
 - terminal logon (VSCS) 643
 - VSCS terminal logon 643
- VSCS and GCS environment (VM) 628, 629
- VSCS printer (VM)
 - logged on to VSCS 642
 - TIMEREL (VSCS option) 642
 - VSCS printer table 642
 - warning, printer problem 643
- VSCS start options (VM)
 - controlling storage management
 - STCHKTM option 633
 - STRELTm option 633
 - personalized 639
 - SCIP recovery 638
 - VSCS dumps, controlling 635
 - VSCS initialization parameters 639
 - VSCS performance, controlling 636
- VSCS VM logo 645
- VSE
 - installing VTAM under 107
 - system generation statements 114
- VTAM Common Network Services (MVS, VM)
 - overview 195
 - X.25 PSDN and (VM) 246
 - XCA (3172) and 195
- VTAM internal trace (VIT) analysis tool
 - data sets 22
 - installing 35, 96
- VTAM restricted materials 695
- VTAM terminal I/O coordinator (VTIOC) 611
- VTAM, defined as
 - CDRM major node 393
 - CDRM minor node 394
 - cross-domain resource manager (CDRM) 393
 - external CDRM 393
 - host CDRM 393
- VTAM, defining to MVS
 - adding channel-attached devices 32
 - DATAMGT macroinstruction 32
 - DATASET macroinstruction 31
 - IODEVICE macroinstruction 32
 - routing code, changes 32
 - SECONSLE macroinstruction 32
- VTAM, gateway (MVS, VM)
 - CDRM definition statement 456
 - defining 445
 - GWPATH definition statement 456
 - NETWORK definition statement 455
- VTAMLST file
 - sample 136

Index

VTAMSEG major node 137
VTIOC (VTAM terminal I/O coordinator) 611

W

warm start (MVS, VSE)
 configuration restart 342
 initial status and 137
warning
 changing messages 353
 exit routine 329
 VSCS, printer problem (VM) 643
window size, adaptive pacing 334

X

X.21 lines (VSE)
 call progress signal retry 238
 direct call 238
 NCP to peripheral node 143, 259
 Short Hold Mode/Multiple Port Sharing 237
X.25 packet switched data network (VM, VSE)
 DTE-to-DTE single-domain connection 247
 ICA connected 243
 optional user facilities 246
 sample 245
 VTAM Common Network Services (VCNS)
 (VM) 246
XCA (external communication adapter)
 APPN connections 539
 connection network, defining a connection to 533
 link-attached NCP support 212
 multiple-domain connections
 CSMA/CD 802.3 network 274
 Fiber Distributed Data Interface network
 (FDDI) 274
 token-ring network 274
 NetView and 231
 single-domain connections
 CSMA/CD 802.3 network 231
 Fiber Distributed Data Interface network
 (FDDI) 231
 token-ring network 231
 tuning statistics 570
 VCNS and (MVS, VM) 195
XDBUF buffer pool
 overview 585
XID operand for an APPN connection 534
XID processing
 type 2.1 PUs 140
 VTAM switched SDLC connection 254
XMITDLY operand on LINE definition statement 249
XNETALS start option 250
XRF (extended recovery facility) (MVS)
 cryptography and 175
 description 174

Communicating Your Comments to IBM

VTAM™
Network Implementation Guide
Version 4 Release 2 for MVS, VM, and VSE
Publication No. SC31-6494-01

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
United States and Canada: **1-800-227-5088**
- If you prefer to send comments electronically, use this network ID:
 - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
 - IBMLink: **CIBMORCF at RALVM13**
 - Internet: **USIB2HPD@VNET.IBM.COM**

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Help us help you!

VTAM™
Network Implementation Guide
Version 4 Release 2 for MVS, VM, and VSE
Publication No. SC31-6494-01

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development
Department E15
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990



Fold and Tape

Please do not staple

Fold and Tape

Help us help you!

VTAM™
Network Implementation Guide
Version 4 Release 2 for MVS, VM, and VSE
Publication No. SC31-6494-01

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name

Address

Company or Organization

Phone No.



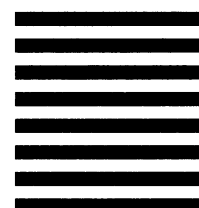
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development
Department E15
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990



Fold and Tape

Please do not staple

Fold and Tape



File Number: S370/4300/30XX-50
Program Number: 5695-117 (MVS/ESA)
5654-010 (VM/ESA)
5686-065 (VSE/ESA)



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-6494-01

